

(19) 日本国特許庁 (J P)

(12) 公表特許公報 (A)

(11) 特許出願公表番号

特表2001-526814

(P2001-526814A)

(43) 公表日 平成13年12月18日 (2001. 12. 18)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード <sup>*</sup> (参考)
G 0 6 F 12/00	5 4 6	G 0 6 F 12/00	5 4 6 K
13/00	5 4 0	13/00	5 4 0 B
15/16	6 4 5	15/16	6 4 5
17/30	1 1 0	17/30	1 1 0 F
H 0 4 L 12/54		H 0 4 L 13/00	3 0 5 Z

審査請求 未請求 予備審査請求 有 (全 66 頁) 最終頁に続く

(21) 出願番号 特願平10-550458  
(86) (22) 出願日 平成10年5月15日 (1998. 5. 15)  
(85) 翻訳文提出日 平成11年11月22日 (1999. 11. 22)  
(86) 国際出願番号 P C T / U S 9 8 / 0 9 9 4 3  
(87) 国際公開番号 W O 9 8 / 5 3 4 1 0  
(87) 国際公開日 平成10年11月26日 (1998. 11. 26)  
(31) 優先権主張番号 0 8 / 8 6 1 , 9 3 4  
(32) 優先日 平成9年5月22日 (1997. 5. 22)  
(33) 優先権主張国 米国 (U S)

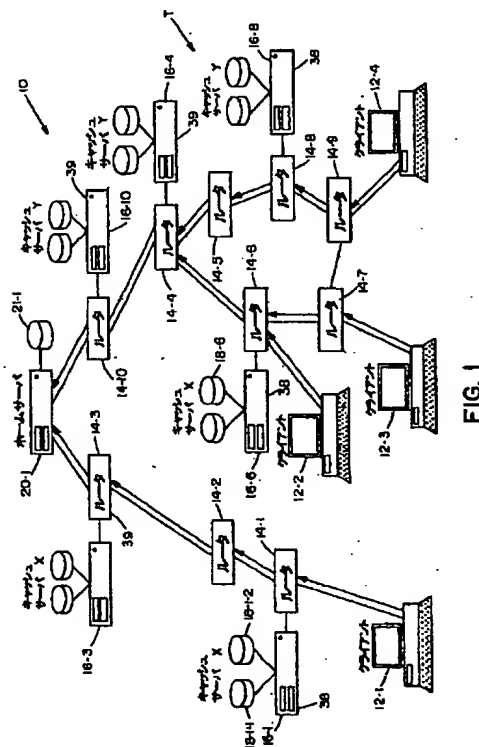
(71) 出願人 トラストィーズ・オブ・ボストン・ユニバ  
ーシティ  
アメリカ合衆国, マサチューセッツ州  
02215, ボストン, ベイ ステート ロー  
ド 147  
(72) 発明者 ミルダット・スライマン・エー  
アメリカ合衆国, マサチューセッツ州  
02169, クインシィ, ハンコック ストリ  
ート 1025, アパートメント 8エッチ  
(74) 代理人 弁理士 杉本 修司 (外2名)

最終頁に続く

(54) 【発明の名称】 分散型キャッシュ、プリフェッチ、複写の方法およびそのシステム

(57) 【要約】

特定ドキュメントに対する要求メッセージが、ルーチンググラフを形成する、クライアントからホームサーバへのパスに沿って、ネットワーク内で自動的、透過的、分散的、・拡大縮小可能および強固なキャッシュ、プリフェッチ、複写をする技術。クライアント要求メッセージは、キャッシュの無い場合は、このルーチンググラフをホームサーバ方向にさかのぼる。しかし、キャッシュサーバがルートに沿って設置されており、サービスが可能であれば、要求をインタセプトする。このような方法で、標準的なプロトコルから逸脱することなく、要求をサービスするために、キャッシュサーバは組み合わされたルータ内にバケットフィルタを装着できる必要があり、またクライアントの見地からホームサーバを代理する必要がある。複数のキャッシュサーバは、ローカル負荷、近接キャッシュの負荷、接続された通信パスの負荷、およびドキュメント人気度に基づいて、ドキュメントをキャッシュおよび廃棄することで、クライアント要求をサービスするために協力する。キャッシュサーバは、また、セキュリティ対策およびドキュメント変換機能を実



**【特許請求の範囲】**

1. 通信プロトコルを使用してネットワークを介して通信する複数のコンピュータであって、ドキュメントの形式で情報を記憶するために前記ネットワーク内の特定ノードにホームサーバまたは単なるサーバとして動作するコンピュータと、ドキュメント要求メッセージをアプリケーション層の前記サーバに送るクライアントとして動作する特定のその他のコンピュータとを有し、そのドキュメント要求メッセージが前記サーバに記憶するドキュメントに対する要求であるシステムにおいて、ドキュメント要求メッセージを実行する方法であって、

(a) 前記ネットワーク内の複数の中間ノード位置にあるドキュメントのローカルキャッシュコピーを記憶し、

(b) 前記サーバの特定の1つに送ろうとする特定のアプリケーション層ドキュメント要求メッセージを発生する前記クライアントの特定の1つに応答して、前記中間ノード位置の1つにおいて、前記アプリケーション層より下位の選択された通信層で前記特定のアプリケーション層ドキュメント要求メッセージを実行して、前記ドキュメント要求メッセージをインタセプトして前記ローカルキャッシュコピーの1つを前記クライアントのアプリケーション層に返送し、前記アプリケーション層ドキュメント要求メッセージが前記中間ノードの下位層でインタセプトされて、前記サーバ上のアプリケーション層がアプリケーション層ドキュメント要求メッセージを受信しないようにする、各ステップを有する方法。

2. 請求項1において、選択された中間ノードにおいて、さらに、

(c) 特定のドキュメント要求メッセージが、前記ローカルキャッシュコピーの1つを提供しても実行できない場合は、そのドキュメント要求メッセージを他のノードにルーティングするステップを有する方法。

3. 請求項1において、前記特定のドキュメント要求を実行するステッ

プがさらに、クライアントでもサーバでも無い最低1つの前記中間ノード内で、

(c) 他の中間ノード上にも記憶されている特定ドキュメントのローカルキャッ

シュコピーを記憶し、

(d) ローカルキャッシュコピーを記憶する近接中間ノードのアイデンティティを決定し、

(e) 前記中間ノードのドキュメント要求メッセージ負荷実行資源の有用性を基に、前記中間ノードと近接中間ノード間でドキュメント要求メッセージの実行を割り当てる、各ステップを有する方法。

4. 請求項3において、前記メッセージ負荷実行資源が前記キャッシュを含む方法。

5. 請求項3において、前記メッセージ負荷実行資源が通信パス負荷を含む方法。

6. 請求項3において、前記ドキュメント要求メッセージの実行を割り当てるステップがさらに、

(f) 処理負荷、通信パス負荷、ドキュメントサイズ、ドキュメント要求メッセージ応答時間、ドキュメント要求メッセージの要求率、ドキュメント要求メッセージの変更率、またはホームサーバ運用性の少なくとも1つから選択された情報を含む状態メッセージを、前記中間ノードと近接ノード間で交換するステップを有する方法。

7. 請求項1において、さらに、

(c) 他の中間ノード上にも記憶されている特定ドキュメントのローカルキャッシュコピーを記憶し、

(d) ローカルキャッシュコピーを記憶する近接中間ノードの前記アイデンティティを決定し、

(e) 前記通信パス負荷を基に、前記中間ノードと近接中間ノード間で前記ドキュメント要求メッセージの実行を割り当てる

各ステップを有することにより、前記特定のドキュメント要求を実行するステップが、前記ノードを相互接続するパス間の通信パス負荷分散を実行する方法。

8. 請求項2において、選択された中間ノードにおいて、さらに、

(d) そこから特定のドキュメント要求メッセージを受け取る第1の近接ノードのアイデンティティを決定し、

(e) 前記特定のドキュメント要求メッセージが前記クライアントに対し前記ローカルキャッシュコピーを返送することによっても実行されない場合は、前記ホームサーバへのパス上で、そこに特定のドキュメント要求メッセージがルーティングされる第2近接ノードのアイデンティティを決定する  
各ステップを有する方法。

9. 請求項8において、ローカルキャッシュコピーを記憶するステップがさらに、

(f) 前記第1近接ノードのノード状態パラメータと前記ローカルノードのノード状態パラメータを決定して、その第1近接ノードとローカルノードのノード状態パラメータが、相互に所定の量だけ異なるとき、前記ローカルキャッシュコピーの少なくとも1つのコピーを前記第1近接ノードに送出して前記第1近接ノード位置に記憶するステップを有する方法。

10. 請求項9において、前記ノード状態パラメータが、要求実行率、要求実行変化、ドキュメントサイズ、ドキュメントフェッチ応答時間、通信パス負荷、キャッシュサーバ負荷、またはホームサーバ動作状態から成るグループから選択される方法。

11. 請求項8において、ローカルキャッシュコピーを記憶するステップがさらに、

(f) 前記第2近接ノードのノード状態パラメータとローカルノードのノード状態パラメータを決定して、前記第2近接ノードとローカルノード

のノード状態パラメータが所定の量だけ異なるとき、前記ローカルキャッシュコピーの少なくとも1つを送出するステップを有する方法。

12. 請求項11において、前記ノード状態パラメータが、要求実行率、要求実行率の変化、ドキュメントサイズ、ドキュメントフェッチ応答時間、通信パス負荷、キャッシュサーバ負荷、またはホームサーバ動作状態から成るグループから選択される方法。

13. 請求項8において、ローカルキャッシュコピーを記憶するステップがさらに、

(f) 前記第2近接ノードのノード状態パラメータとローカルノードのノード状態パラメータを決定して、前記第2近接ノードとローカルノードのノード状態パラメータが所定の量だけ異なるとき、前記フィルタリングのステップにおいて、前記クライアントにローカルキャッシュコピーを提供して実行される前記ドキュメント要求の部分を減少するステップを有する方法。

14. 請求項1において、前記フィルタリングのステップがさらに、

(c) ローカルノード状態パラメータが所定の量と異なる場合は、前記クライアントにローカルキャッシュコピーを返送するステップを有する方法。

15. 請求項1において、前記の選択された中間ノードにおいて、さらに、

(c) 前記ローカルキャッシュコピーの1つを前記クライアントに提供して実行される要求メッセージの数に関する負荷統計を記録するステップを有する方法。

16. 請求項1において、選択された中間ノードにおいて、さらに、

(c) 特定のドキュメントに対して受け取られて実行された要求メッセージの数に関する要求メッセージ統計を記録するステップを有する方法。

17. 請求項11において、前記の選択された中間ノードにおいて、さ

らに、

(c) 特定のサーバに記憶されたドキュメントに対して受け取られた要求メッセージの数に関する応答時間統計を記録するステップを有する方法。

18. 請求項1において、ドキュメント要求メッセージが複数の通信パス上の前記中間ノードで受け取られ、さらに、前記の選択された中間ノードにおいて、

(c) ドキュメント要求メッセージが特定のパスから受け取られた記録に関する要求メッセージ統計を記録するステップを有する方法。

19. 請求項1において、ドキュメント要求メッセージが、複数の通信パス上の前記中間ノードで受け取られ、さらに、全体のネットワーク需要とネットワーク領域需要の1つから選択された基準によるか、または特定のクライアント需要によって、通信パスの使用頻度を予測する方法。

20. 請求項1において、前記ドキュメント要求メッセージをフィルタリングするステップがさらに、

(c) 前記ドキュメント要求メッセージドを送出する前に、それを介してドキュメント要求メッセージがクライアントから届くノードを認証するステップを有する方法。

21. 請求項1において、前記ドキュメント要求メッセージをフィルタリングするステップがさらに、

(c) 前記ローカルキャッシュコピーを前記クライアントに返送する前に、前記のキャッシュされたドキュメントが発生したノードを認証するステップを有する方法。

22. 請求項1において、前記サーバが前記ドキュメント上で動作するプログラムも記憶し、ステップ(b)が、前記中間ノードにおいて、さらに、

(c) 要求されたドキュメントコピーに関する特定の選択されたプロ

グラムが前記サーバから得られるように、選択されたプログラムのローカルキャッシュコピーを記憶するステップを有する方法。

23. 請求項22において、さらに、

(d) 前記クライアントからの要求に応じて、前記中間ノードにおいて選択されたプログラムの前記ローカルキャッシュコピーを実行するステップを有する方法。

24. 請求項22において、さらに、

(d) 前記中間ノードと前記クライアント間の相互作用的セッションを維持して、前記サーバとして動作する前記中間ノードを、前記サーバが前記ドキュメント要求メッセージを実行した場合に動作させるステップを有する方法。

25. 請求項1において、前記ドキュメント要求メッセージをフィルタリングするステップがさらに、

(c) 選択されたプログラムを前記ローカルキャッシュコピーに適用して結果を得て、その後そのプログラムの適用結果をクライアントに返送するステップを有する方法。

26. 請求項1において、前記ドキュメントがマルチメディアドキュメント、プログラム、データベース、圧縮データ、または暗号化データのグループから選択される方法。

27. 請求項8において、複数の中間ノードが前記サーバと前記クライアント間に設置され、前記ドキュメントの需要予測率に基づいて、特定ドキュメントが複数のノードに対してルーチングされてネットワークに送出される方法。

28. 請求項9において、前記所定の要求実行率が、予測要求実行率、予測要求実行率変化、ドキュメントサイズ、予測ドキュメントフェッチ応答時間、または予測キャッシュサーバ負荷のグループから選択されたドキュメント属性に依存する方法。

29. 請求項1において、前記中間ノードにおいて、さらに、

(c) 前記ローカルキャッシュコピーと共に、前記ローカルキャッシュコピーの放出についての条件を示すデータを記憶し、

(d) 前記条件を満足した場合にだけ、前記ローカルキャッシュコピーを前記クライアントに返送する  
各ステップを有する方法。

30. 請求項29において、前記条件が前記ドキュメントを放出した時間である方法。

31. 請求項1において、キャッシュコピーを記憶するステップが、所定の基準条件を基に選択的に実行される方法。

32. 請求項31において、前記条件が時刻を含む方法。

33. 請求項31において、前記条件基準がドキュメントサイズ、希望されるドキュメント要求メッセージ応答時間、ドキュメント要求メッセージ率、ドキュメント要求メッセージ率の変化率、サーバ負荷、通信パス負荷、またはホームサーバ動作状態のグループから選択された少なくとも1つである方法。

34. 請求項1において、キャッシュコピーを記憶するステップがさらに、所定のサイズ以上のドキュメントの部分コピーを記憶するステップを有する方法。

35. 請求項1において、さらに、

(c) 所定の基準条件を基に実行されたドキュメントのキャッシュコピーを選択的に消去するステップを有する方法。

36. 請求項35において、前記所定の基準条件が時刻を含む方法。

37. 請求項35において、前記所定の基準条件がドキュメントサイズ、希望されるドキュメント要求メッセージ応答時間、ドキュメント要求メッセージ率、ドキュメント要求メッセージ率の変化率、サーバ負荷、または通信パス負荷のグループから選択された少なくとも1つである方法。

38. 請求項1において、キャッシュコピーを記憶するステップがさら

に、所定のサイズ以上のドキュメントの部分コピーを記憶するステップを有する方法。

39. コンピュータが複数のノードにおいて複数のパスにより相互接続されるコンピュータネットワークで動作する論理的通信ネットワークを提供する方法であって、その論理的通信ネットワークにより第1ネットワークの第1コンピュータが確実に第2ネットワークノードの第2コンピュータと通信できるようにするもので、

(a) 前記第1ノードと、前記第1ノードと前記第2ノード間の通信パス内の第1ホップである第1の中間ノードとの間のパスに沿った確実な通信パスを提供し、

(b) 前記第2ノードと、前記第1ノードと前記第2ノード間のパス内の最終ホップである最終の中間ノードとの間の確実な通信パスを提供し、

(c) 前記第1および最終中間ノードに対し暗号化鍵を提供し、前記第1中間ノードがその鍵を使用して情報を暗号化した後に前記パス内の他の中間ノードに送出し、前記最終ノードがその鍵を使用して情報を暗号化した後に前記第2中間ノードに送出して、その結果、確実な通信パスが第1と最終中間ノード間に提供される

各ステップを有する方法。

40. コンピュータが複数のノードにおいて複数のパスにより相互接続されるコンピュータネットワークで動作する通信ネットワークを提供する方法であって、



その通信ネットワークにより第1のネットワークノードの第1コンピュータが第2ネットワークノードの第2コンピュータと通信できるようにするもので、

(a) 前記第1ノードと、前記第1ノードと前記第2ノード間の通信パス内の第1ホップである第1の中間ノードとの間のパスに沿った通信パスであって、複数のパスを有する通信パスを提供し、

(b) 前記第2ノードと、前記第1ノードと前記第2ノード間のパス内

の最終ホップである最終の中間ノードとの間の通信パスを提供し、

(c) 前記第1および最終中間ノードに対し圧縮と復元の機能を提供し、前記第1中間ノードがその圧縮機能を使用して情報を圧縮した後に前記パス内の他の中間ノードに送出し、前記最終ノードが前記復元機能を使用して情報を復元した後に前記第2中間ノードに送出して、その結果、帯域幅を縮小した通信パスが前記第1と第2の中間ノード間に提供される

各ステップを有する方法。

41. 請求項1において、キャッシュの一貫性を維持するために、さらに、

(c) 前記サーバにおいて、前記ドキュメントにドキュメント終了タイムスタンプを付し、

(d) 中間ノードにおいて、キャッシュされたドキュメントに付された前記タイムスタンプを検査して、ドキュメントが終了したかどうか判定し、

(e) 前記キャッシュされたドキュメントが終了していた場合は、そのドキュメントを消去または再生する

各ステップを有する方法。

42. 請求項1において、さらに、

(c) 前記サーバにおいて、前記ドキュメントにドキュメント変更タイムスタンプを付し、

(d) 中間ノードにおいて、前記ドキュメントの変更率を評価して、そのドキュメント変更率が前記要求率よりも所定量大きい場合は、前記キャッシュされたコピーを消去し、そのドキュメント変更率が前記要求率よりも所定量小さい場合は、キャッシュコピーの更新を要求する

各ステップを有する方法。

43. 階層化通信プロトコルを使用してネットワークを介して通信する複数のコンピュータを含むシステムにおいて、そのネットワーク内の特

定のノードのコンピュータがドキュメントの形式で情報を記憶するサーバとして動作し、特定の他のコンピュータがドキュメント要求メッセージをアプリケーション層レベルの前記サーバに送るクライアントとして動作し、前記ドキュメント要求メッセージが前記サーバに記憶されたドキュメントに対する要求であって、透過的な代理によってドキュメント要求メッセージを実行する方法であって、

(a) ネットワーク内の複数の中間ノード位置にローカルキャッシュコピーを記憶し、

(b) 前記サーバの特定の1つに送ろうとする特定のアプリケーション層ドキュメント要求メッセージを発生する前記クライアントの特定の1つに応答して、そのドキュメント要求メッセージをインタセプトして前記中間ノード位置の1つで前記アプリケーション層ドキュメント要求メッセージを実行して、前記ローカルキャッシュコピーの1つをクライアントのアプリケーション層に返送し、その結果前記アプリケーション層要求メッセージが中間ノードでインタセプトされて、その結果ネットワーク接続が前記サーバのアプリケーション層との間で確立されないようにする

各ステップを有する方法。

44. 請求項43において、中間ノードがキャッシュサーバとルータとを有し、そのルータにおいて、さらに、

(c) インタセプトされるドキュメント要求メッセージを認識して、そのメッセージを抜き出して前記キャッシュサーバで処理するステップを有する方法。

45. 請求項1において、さらに、

(c) 最も頻繁に連続して要求される関係ドキュメントを合わせて同時にプリフェッチするステップを有する方法。

46. 請求項1において、さらに、

前記クライアントにおいて、

(c) 前記クライアントとサーバ間に通信パスを確立することを要求する接続要求メッセージを発生し、

中間ノードにおいて、

(d) 前記クライアントから接続要求メッセージを受け取り次第、ドキュメント要求メッセージの受信を待ち、その接続要求メッセージと前記ドキュメント要求メッセージを合わせて前記パス内の次の中間ノードに送出する

各ステップを有する方法。

47. 請求項46において、さらに、前記要求メッセージにより指示された前記ドキュメントをキャッシュする中間ノードにおいて、

(e) 前記クライアントに対し前記接続要求を確認応答し、そのクライアントに前記の要求されたドキュメントを返送するステップを有する方法。

48. 請求項1において、さらに、

(c) ローカルキャッシュに記憶するドキュメントとそのローカルキャッシュから取り除くドキュメントを選択するキャッシュサーバを動作させて、そのキャッシュサーバに最も頻繁に要求されるドキュメントを選択してパス内のノードの近接キャッシュサーバに複写させ、ほとんど要求されないドキュメントをメモリから削除させるステップを有する方法。

49. 請求項1において、各中間ノードにはリソースマネージャが組み合わされており、このリソースマネージャが、

(c) 設計上で重要と考えられているメッセージトラフィックを向上させるために、入力および出力メッセージを緩衝記憶する通信バッファの使用を割り当てるステップを遂行する方法。

50. 請求項49において、

(d) ドキュメントサイズ、ドキュメント種類、伝送方向、または優先度の組み合わせから選択された内容属性を基に、通信バッファおよび通

信パス帯域幅の使用を割り当てるステップを有する方法。

51. 請求項1において、前記中間ノードにおいて、さらに、

(c) 前記中間ノードのノード状態パラメータを近接サーバのノード状態パラメ

ータと比較し、

(d) 前記近接ノードのパラメータおよび前記中間ノードパラメータが、所定の時間長さ以上に所定量だけ異なる場合は、前記中間ノードにおいて、その近接ノードが中間ノードで要求された1つまたは複数のドキュメントをキャッシュしないと推定する

各ステップを有する方法。

52. 請求項51において、前記ノード状態パラメータが、要求実行率、要求実行率の変化、ドキュメントサイズ、ドキュメントフェッチ応答時間、通信パス負荷、ホームサーバ運用性、またはキャッシュサーバ負荷から成るグループから選択される方法。

53. ネットワークを介して通信する複数のコンピュータを含み、そのネットワーク内の特定のノードのコンピュータがドキュメントの形式で情報を記憶するサーバとして動作し、特定の他のコンピュータがドキュメントを前記サーバに記憶されるように要求するドキュメント要求メッセージを送るクライアントとして動作するシステムにおいて、ドキュメント要求メッセージの実行を調整して負荷を分散する方法であって、

(a) ネットワーク内の複数の中間ノード位置にローカルキャッシュコピーを記憶するステップと、

前記サーバの特定の1つに送ろうとする特定のドキュメント要求メッセージを発生する前記クライアントの特定の1つに応答して、

(b) その特定ドキュメント要求メッセージ内で指定されたドキュメントに対応する前記キャッシュサーバの1つを返送して、複数の中間ノード位置の1つで前記特定ドキュメント要求メッセージを実行して、前記ローカルキャッシュコピーの1つをクライアントのアプリケーション層に返送し、その結果、前記中間ノード位置のアプリケーション層がその

要求をインタセプトして前記サーバのアプリケーション層が前記ドキュメント要求メッセージを受け取らないようにするステップと  
を有する方法。

54. 請求項53において、さらに、

(c) 前記複数の中間ノード位置間で、前記キャッシュコピーを記憶するステップを調整し、かつ、前記複数の中間ノード位置間でドキュメント実行を分割して、前記中間ノードキャッシュサーバまたはホームサーバにドキュメント要求負荷を分散するステップを有する方法。

55. 請求項53において、さらに、

(c) 前記中間ノード位置において、キャッシュコピーを記憶するステップを調整し、かつ、前記複数の中間ノード位置間でドキュメント実行を分割して、通信パス負荷を分散するステップを有する方法。

56. 請求項53において、キャッシュコピーを記憶するステップを前記複数の中間ノード位置間で調整してドキュメント負荷を減少させるステップを調整するステップが、

(c) 近接するノード位置においてドキュメントを複写するステップを有する方法。

57. 請求項53において、中間ノード位置にドキュメントのコピーを記憶するステップが、さらに、

(c) 所定の基準に基づいて、そのようなコピーを有効または無効のどちらかにすることで、ドキュメント要求を実行するステップを有する方法。

58. 請求項57において、前記所定の基準が時刻である方法。

59. 請求項53において、前記中間サーバの1つで前記特定ドキュメント要求メッセージを実行するステップが、さらに、

(c) 前記クライアントからドキュメント要求メッセージを最初に受け取る前記中間ノード位置の1つであるリーフノードと、前記サーバとの間の通信接続をオープンし、

(d) 前記ドキュメント要求および前記通信接続を、前記リーフノードよりもネットワーク内の前記サーバにより近い他の中間ノードに送出する各ステップを有する方法。

60. 請求項53において、前記中間サーバの1つで前記特定ドキュメント要求

メッセージを実行するステップが、さらに、

(c) 前記クライアントからドキュメント要求メッセージを最初に受け取る前記中間ノード位置の1つにあるリーフノードと、前記サーバとの間の通信接続をオープンし、

(d) 前記ドキュメント要求を、前記リーフノードよりもネットワーク内の前記サーバにより近い他の中間ノードに中継する

各ステップを有する方法。

61. 請求項53において、前記キャッシュサーバが、さらに、

(c) 前記クライアントの観点から前記ホームサーバに対して通信代理として動作するステップを遂行する方法。

62. 請求項53において、前記ドキュメント要求メッセージを実行するステップがさらにドキュメントへのアクセスを制御するものであり、さらに、

(c) 前記ドキュメント要求メッセージ内のURL (Uniform Resource Location) フィールドに基づき、ドキュメント要求メッセージをフィルタリングするステップを有する方法。

63. 請求項53において、前記ドキュメント要求メッセージを実行するステップがさらにドキュメントへのアクセスを制御するものであり、さらに、

(c) 前記ドキュメント要求メッセージ内の確認フィールドに基づき、ドキュメント要求メッセージをフィルタリングするステップを有する方法。

64. 請求項53において、前記クライアントに前記ドキュメントのキ

ャッシュコピーを返送するステップが、さらに、

(c) 前記ドキュメントを精細に検査するステップを有する方法。

65. 請求項53において、前記クライアントに前記ドキュメントのキャッシュコピーを返送するステップが、さらに、

(c) 前記ドキュメントをコード認定するステップを有する方法。

66. 請求項53において、前記クライアントに前記ドキュメントのキャッシュコピーを返送するステップが、さらに、

(c) 前記ドキュメントを解読するステップを有する方法。

67. 請求項53において、前記クライアントに前記ドキュメントのキャッシュコピーを返送するステップが、さらに、

(c) クライアント認証およびホームサーバ要求に基づき、前記ドキュメントのキャッシュコピーへのアクセスを制御するステップを有する方法。

68. コンピュータネットワーク内の情報に対する要求を実行する方法であって、少なくともそのネットワーク内のいくつかコンピュータが情報を記憶するホームサーバとして動作し、少なくともいくつかコンピュータが前記ホームサーバに要求メッセージを送るクライアントとして動作するもので、

(a) キャッシュサーバとして動作する前記ネットワーク内の複数のコンピュータ内に情報のコピーを記憶することにより、そのネットワークを介して情報のキャッシュコピーを分散し、

(b) 前記ネットワーク内の複数の中間コンピュータを通して、前記クライアントから前記サーバに要求メッセージをルーティングし、その際、少なくとも前記中間コンピュータのいくつかがキャッシュサーバとしても動作し、前記要求メッセージが前記サーバの特定の1つから発行されて前記サーバの特定の1つから情報を取得し、

(c) 前記キャッシュサーバを通してルーティングされた通りに要求メッセージを透過的に処理することで、前記ホームサーバの代わりに前記キ

ャッシュサーバから提供される要求メッセージが、前記クライアントとホームサーバに対して透過的な方法で、前記キャッシュサーバから提供されるようにする各ステップを有する方法。

69. 請求項59において、さらに、

(d) 前記キャッシュサーバによる前記要求メッセージの提供に関する所定の基準に対応して、ネットワーク内の前記キャッシュサーバ間でキャッシュコピーを自動的に移動させるステップを有する方法。

70. 請求項1において、前記マルチメディアドキュメントがテキスト、グラフィックス、オーディオ、ビデオ、プログラムまたはその他のデータから選択されたデジタル化情報を含む方法。

71. 請求項1において、前記ネットワークがワイヤレスネットワークを有する方法。

72. 請求項1において、前記中間ノードが、特定の通信エンティティが故障しているかどうか判断し、故障していれば他のネットワークエンティティに通告するステップを遂行する方法。

73. 請求項72において、前記通信エンティティがルータまたは通信パスのいずれか1つである方法。

74. 請求項72において、前記他のネットワークエンティティが他の中間ノード、ネットワーク管理者、または他のネットワーク化コンピュータシステムのいずれか1つである方法。

75. 請求項57において、前記所定の基準が前記クライアントによる通告である方法。

76. 請求項57において、前記所定の基準が前記ホームサーバによる通告である方法。

77. 請求項2において、選択された中間ノードにおいて、さらに、

(d) 特定のドキュメント要求メッセージを受け渡す第1近接ノードのアイデン

ティティを決定し、

(e) 前記ホームサーバへのパス上で、特定のドキュメント要求メッセージがルーティングされる第2近接ノードのアイデンティティを決定し、

(f) 前記中間ノードにおいて処理に失敗するかまたは処理が遅いために、前記特定ドキュメント要求メッセージが前記中間ノードで満足に実行されない場合、前記パスを変更してその中間ノードを迂回する各ステップを有する方法。



**【発明の詳細な説明】**

## 分散型キャッシュ、プリフェッチ、複写の方法およびそのシステム

## 発明の背景

インターネット、私用イントラネット、エクストラネット、仮想専用ネットワークのようなコンピュータネットワークは、情報の記憶や再生、通信、電子商取引、エンターテインメントおよびその他の用途を含むさまざまな分野に、ますます多く使用されるようになってきている。これらのネットワークでは、サーバとして知られる特定のコンピュータが情報の記憶と供給のために使用されている。ホストまたはホームサーバとして知られるサーバの一種は、本明細書で「ドキュメント」と称するものを除いて、各種コンピュータファイルフォーマットで保存されたデータやプログラムのような情報にアクセスできる機能を備えている。インターネットでは、ドキュメントは通常、基本的にテキストとグラフィックスから構成されているのに対して、各ドキュメントは、実際には、テキスト、テーブル、グラフィックイメージ、サウンド、動画、アニメーション、コンピュータプログラムコードおよび／またはその他の多くのデジタル化された情報を有する各種の情報を収容するデータ構造を含む高度にフォーマットされたコンピュータファイルである。

ネットワークにおいてクライアントとして知られるその他のコンピュータでは、ユーザがホームサーバに対して、ネットワークを介してコピーをクライアントに送るように要求することにより、ドキュメントにアクセスできる。クライアントがホームサーバから情報を得るために、通常、各ドキュメントは参照できるアドレスを有している。たとえば、インターネットのコンテキストおよびHTTP（ハイパーテキスト転送プロトコル(Hyper Text Transfer Protocol)）として知られる通信プロトコル内において、通常アドレスは、URL (Uniform Resource Locator)として知られる文字数字の列であり、(a) ホームサーバから情報を取得

するために、名前または数字アドレスの形式でホームサーバのアドレスを指定し、さらに、(b) クライアントによって要求された情報を識別するローカル情報

テキスト列（ファイル名、検索要求またはその他の識別情報）を指定する。

ユーザがクライアントコンピュータに対してURLを指定すると、URLのアドレス部分は、ネットワークを介してDNS（ドメイン名サービス(Domain Name Service)）のようなネーミングサービスに送信されて、適正なホームサーバとの接続を確立する方法の命令を取得する。サーバとの接続が確立すると、ローカル情報テキスト列をネットワークを介して直接ホームサーバに送ることにより、クライアントは所望するドキュメントを再生できる。その際、サーバはローカルディスクまたはメモリ記憶からドキュメントを再生して、ネットワークを介してクライアントに伝送する。その後、ホームサーバとクライアント間のネットワーク接続は終了する。

コンピュータおよびネットワーク産業のアナリストや専門家は、現在では、インターネット上の通信量が過大になってきているために、インターネットを使用可能にするための方法が本質的に変わる可能性があることを危惧している。特に、個人の多くは現在、インターネットが耐えられないほどに遅く、もはや現状に適合した情報交換のための信頼できる実体ではないと考えている。

現在のボトルネックは、疑いもなく、ユーザ数のほかに、マルチメディアファイルのような複雑なドキュメントの送信数が、指数関数的に増加していることによる。解決手段は、単に、サーバとクライアント間の物理的接続に対してさらに帯域幅を追加することに見えるかもしれない。しかし、これは、家庭内や世界中の地域に、同軸ケーブルや光ファイバケーブルおよび組み合わせられたモデム等のような広帯域幅の相互接続用ハードウェアを設置する費用を負担してはじめて実現できる。

さらに、帯域幅を追加しても、性能を改善するという保証はない。特

に、ビデオエンターテイメント用のような大容量のマルチメディアファイルが、共通電子メールのようなさらに優先度の高い種類のデータに置き換わる可能性がある。残念ながら、帯域幅配分方式を、既存のネットワーク通信プロトコルを変更せずに実現することは困難である。インターネットで使用されている通信技術はTCP/IPと呼ばれ、単純で明快なプロトコルであり、アップル社のマッキ

ントッシュ (Macintosh) や IBM互換の PC、UNIXワークステーションのような異なる種類の多くのコンピュータでデータを共有して実行できる。TCP/IP プロトコルをアドレスがパケット内容の情報を含むことができるように拡張するという大がかりな提案があるが、これらの提案は技術的に複雑で、かつ膨大なコンピュータネットワークの使用者間の協調を必要とする。このように、既存の TCP/IP プロトコルに変更を加えることを期待するのは非現実的である。

取られた対策は、インターネットの急速に成長している用途が、サーバ容量および通信メディアの帯域幅容量を上回り続けることを認識することであった。これらの方法は、基本的なクライアント/サーバ方式（クライアントが直接ホームサーバに接続されている）が、特に1台のホームサーバから複数のクライアントに情報を広く分配する必要のある場合には資源を浪費することを前提に始まっている。実際に、予期しない要求を受けて対処できないために、インターネットサーバが簡単に故障した例は多い。

ホームサーバに対する要求を軽減するために、大容量の中央ドキュメントキャッシュが使用される。キャッシュは、多くのクライアントから特定のホームサーバに対して、同一ドキュメントを繰り返し要求する無駄を減少させる。類似の要求をインタセプト（横取り）することにより、キャッシュは、複数のクライアントに対して同一ドキュメントのコピーを提供するの用にられる。

クライアント側から見ると、キャッシュとの相互作用は、通常ユーザ

には意識されない方法で行われる。しかし、ネットワークメッセージの観点からは、僅かに異なるものである。その差異は、ホームサーバ接続に必要な情報を索引するために、ドメイン名サービス (DNS) に対して要求のアドレス部分が提供されるときに、DNS が実際の本来のホームサーバに代わってキャッシュのアドレスを返送するようにプログラムされていることである。

反対に、サーバノードは、クライアントに対して代理として動作し、キャッシュコピーを検索するための探査メッセージを送出する。ネットワーク内の特定ノードにおいてキャッシュコピーがいったん発見されると、要求がそのノードに転

送される。たとえば、全米科学財団 (National Science Foundation) の援助で、米国内のさまざまな場所にドキュメントキャッシュが設置された結果、海洋横断ネットワーク接続におけるボトルネックは解消された。一般に、西海岸に設置されたこれらのキャッシュのあるものは、アジア-太平洋および南米諸国からのドキュメント要求を処理しており、東海岸に設置されたものはヨーロッパからのドキュメント要求を処理している。これらの全国的なキャッシュのその他のものは、米国全体に設置された公共的ドキュメントの要求を処理している。

しかし、このようなキャッシュ技術は、必ずしもまたは一般的にもドキュメント要求負荷の最適な分配を達成していない。特に、キャッシュが最も効率的であるためには、DNS ネームサービスまたはその他のメッセージルーチング機構（経路選択機構）が、予測される人気が高いドキュメントに対する要求を適正にインタセプト（横取り）するように修正される必要がある。このようにキャッシュコピーを導入すると、一時的なコピーを置く必要があるため、ネーム解決の通信オーバーヘッドが増加する。ネームサービスは、これらのコピーを実在するように登録し、ドキュメントに対する要求を配布するために情報を広め、適時削除されたキャッシュコピーを記録から確実に除去する必要がある。しばしば

、手動操作によって、キャッシュの検索順を調整し、さらに／またはドキュメント分散の変更を実行する必要がある。

残念ながら、要求パターンが頻繁かつ著しく変わると、キャッシュコピーの識別、位置、数さえも頻繁に変えさせることになる。キャッシュディレクトリの更新を必要とする結果、通常キャッシュコピーは大幅には効率よく複製できないこととなり、ネームサービス自体がボトルネックとなる。

キャッシュを実現する別の可能な方法は、クライアント／サーバの相互作用プロトコルを、完全な分散形プロトコルを使用して、たとえばランダムな方向に探查を発行することにより、クライアントが前もって適切なキャッシュコピーを特定するように、変更することである。既存のプロトコルの修正の複雑さや、このような方法により生じるメッセージコストは別として、このような方法は、全体のドキュメントサービス待ち時間に1つ以上の往復遅延を付加（クライアントに

もそれと分かる) することになる。

#### 発明の概要

本発明は、自動的、分散的、透過的なキャッシュ方法であって、ドキュメント要求が流れるパス、つまり、コンピュータネットワークを通してクライアントから特定のホームサーバ上の特定のドキュメントに続く経路(パス)が、自然にルーチンググラフまたはツリーを形成する事実を利用している。

本発明によれば、キャッシュサーバをネットワーク全体に設置して、ドキュメント要求がルーチンググラフに沿って中間ノードで実行できるなら、中間ノードがクライアントにキャッシュドキュメントを返送するようにする。このようにして、ドキュメント要求メッセージはホームサーバに到達する前に返答される。ドキュメント要求メッセージは、キャッシュが存在しない場合と同様の方法で、クライアントからルーチング

グラフ上方にホームサーバ方向にルーチングできるので、ネームサービスを変更する必要はない。

標準的なネットワークプロトコルから逸脱せずに本方法で要求をサービスできるようにするために、キャッシュサーバは、組み合わせたルータ内にパケットフィルタを有する。このフィルタは、組み合わせたキャッシュ内に高い確率でヒットし易い(キャッシュ内に存在する確率が高い)ドキュメント要求パケットを抜き出す。

好ましくは、キャッシュサーバは、また、ホームサーバの通信プロトコルにおける代理を果たすように動作する。つまり、中間ノード位置においてドキュメント要求メッセージを実行する一部分として、使用中の通信プロトコルに従ってクライアントに適切なメッセージを送り、ドキュメントを実際にホームサーバから受け取ったようにクライアントを信じ込ませる。

本発明は、また、複数のキャッシュサービスがクライアント要求をサービスするのに協力する方法を提供する。特に、各サーバは、そのローカルな負荷、近接キャッシュ上の負荷、隣接通信パスの負荷およびドキュメントの人気度合いに基づいて、ドキュメントをキャッシュおよび廃棄する機能を有する。たとえば、各

サーバは、近接キャッシュの負荷評価を保持し、そのサーバ自体の負荷評価を近接キャッシュに通信する。あるキャッシュサーバが、近接キャッシュのいずれに比べても過負荷であると認識すると、その仕事の一部を軽減するか、軽負荷の近接サーバに転送する。これを実行するために、好ましくは、キャッシュサーバは、また、所定のホームサーバに対する経路を示すルーチンググラフ上の近接キャッシュの上流（つまり親）および下流（つまり子）のノードの識別を覚える。

本発明にかかるドキュメントキャッシュシステムの基本概念にはいくつかの利点がある。

第1に、この方法は、キャッシュディレクトリからのアドレス検索要

求、ドキュメント要求のリダイレクト、またはキャッシュコピーを配置するためのネットワークの別の構成要素の探査を必要としないことである。したがって、キャッシュコピーの存在場所は、要求メッセージが流れる自然な経路に沿って、偶然的に決まる。これにより、クライアントは、適切なキャッシュコピーを見つけるためにネットワークの別のエンティティを待つのに関連する遅延やボトルネックを回避できる。

さらに、システム全体として、必要ならネットワークを通してドキュメントのキャッシュコピーを拡散でき、その代わりにキャッシュおよび通信経路に沿ったボトルネックを拡散できる。

また、キャッシュコピーは常にクライアントよりも本来のサーバ近くに置かれるため、ネットワーク帯域幅の浪費を抑え応答時間を短縮できる。したがって、ドキュメント要求メッセージおよびドキュメント自体は、通常、要求される度にサーバと各クライアント間の全距離を移動する必要がある。このようにして、全体のネットワーク帯域幅は節約され、応答時間は短縮され、負荷は全体的に平均化する。

このように、本発明は、既存のネットワークプロトコルを修正することなく、ホストサーバ上の負荷需要における差異を抑えるのに役立つだけでなく、ネットワーク通信資源上の負荷を低減する。

さらに、キャッシュコピーがネットワーク上に分散されているため、キャッシ

システムは故障に強く安全確実になる。  
。

またこの技術は、キャッシュサーバをさらに追加することで、クライアントとサーバの両方に一層の便宜をもたらすという意味で、規模の拡大縮小が可能である。

本発明は、ネットワークに付加的な機能を実現するのに使用できる。これら各種機能は、キャッシュサーバに組み合わせたルータに実装されたパケットフィルタも、また、ローカルキャッシュに対するドキュメントのコピーの簡単な派生的要求以上のことができる、という事実からく

る。

たとえば、キャッシュするドキュメントの選択を管理するためには、人気度の統計をキャッシュサーバにより収集する必要がある。このため、各キャッシュサーバは、特定のドキュメントに対する参照回数を受信し、どこから受信したかの記録を取り、総計の要求量と要求率を求めることができる。このデータはネットワーク領域ごとに中央に集計され、ドキュメント発行元は、彼らの資料（提供物）上へのヒット率だけでなく、ネットワークのどの場所からヒットがなされたのかの統計を得ることが可能となる。これは、ドキュメント人気の分析だけでなく、電子商取引や知的財産追跡に対しても重要である。

キャッシュサーバは、また、サービスプロバイダの見地から、負荷を分割するように動作させて、データベースやサーチエンジンインデックスファイル等のような人気のあるドキュメントの複製を管理するのに使用できる。すなわち、データデースプロバイダは、ドキュメントを最善の配置になるようにネットワーク内のどこにでも置くことができ、そのドキュメントにアクセスするクライアントにより近いデータを送り出す。

セキュリティ構成セットをキャッシュサーバに付加することも容易である。

そのような構成の1つは、要求メッセージとその他の情報の発信元の認証である。これは、キャッシュサーバが、ドキュメント要求メッセージおよびドキュメント自体の物理的発信源に対する情報を保持しているために可能となる。そのメ

カニズムは、また、各ノードがフィルタとパケットルータを有する事実による。フィルタとパケットルータは、要求をキャッシュコピーにリダイレクトする方法の記録だけでなく、情報に対する要求を認証してキャッシュコピーへのアクセスを制限するのに使用される。本発明は、また、ネットワークにおける各種の方式のドキュメント分散を可能とする。たとえば、本発明はドキュメント圧縮（別の

形で帯域幅を節約する）や暗号化、特定のサーバおよびクライアントノードがキャッシュサーバを使用して通信をコミットされている限り（たとえば、ネットワーク内のクライアントとサーバ間のパスに沿った最初と最後のノードがキャッシュサーバを含む）可能である。

本発明は、また、動的内容のドキュメントを効率良くキャッシュできるようにする。このような動的内容のドキュメントは、通常プログラム命令に従って、実行中にもクライアント変更に応答する方式である。本発明は、キャッシュ内に動的内容のドキュメントの存在を認識すると、ドキュメント用データをキャッシュするだけでなく、ドキュメントの表示方法またはデータを再生する時を指定するプログラムをフェッチして実行できるようにする。

本発明によれば、また、サーバとクライアント間のネットワークノード数が減少するので、オーディオやビデオデータファイルのような記憶されている連続的メディアの配布も効率よくできる。

#### 図面の簡単な説明

本発明による利点をさらに理解するためには、添付図面と合わせて次の説明を参照されたい。

図1は、代表的なコンピュータネットワークを示す図であって、単一ドキュメントに対する要求経路および本発明による経路に沿ったキャッシュサーバ位置を示す。

図2は、リソースマネージャ、プロトコルプロキシおよびスヌーパが、どのように本発明の実現に使用されているかを示す通信階層図である。

図3は、ネットワーク上におけるドキュメント要求の代表的な行程を示す図である。



図 4 は、本発明によるルーチング経路上に配置かれたリーフサーバにより実行される動作のフローチャートである。

図 5 は、中間ノンリーフキャッシュサーバにより実行される動作のフローチャートである。

図 6 は、ルーチング経路上の最終キャッシュサーバで実行される動作のフローチャートである。

図 7 は、中間サーバによるドキュメント要求メッセージのインタセプトを示す図である。

図 8 は、さらに詳細にドキュメント要求メッセージのインタセプトを示す図である。

図 9 (a) および 9 (b) は、最悪例のクライアント要求の際に、どのように拡散が進行するかを示す図である。

図 10 は、キャッシュサーバがドキュメント変換機能を実行する方法を示す図である。

#### 好ましい実施形態の詳細説明

##### 1. はじめに

図 1 に示すように、インターネット、エクストラネット、私用イントラネット、仮想専用ネットワーク、ローカルエリアネットワークまたはその他の種類のコンピュータネットワークのようなコンピュータネットワーク 10 は、クライアントコンピュータ 12-1、12-2、12-3、...、12-4 (まとめて、クライアント 12)、ルータ 14-1、14-2、...、14-10、キャッシュサーバ 16-1、16-3、16-4、16-6、16-8、16-10 およびホームサーバ 20 を有する多数のネットワークエンティティから構成されている。ネットワークは、公共および私設の電話線、マイクロ波リンク、セルラーおよび無線、衛星リンク、その他の種類のデータ伝送のような、任意および各種の物理層信号伝送メディアを使用できる。

図示したネットワークでは、特定のルータ 14 はキャッシュサーバ 16 とそれぞれ組み合わされている。一方、その他のルータはキャッシュ

サーバと組み合わせられていない。キャッシュサーバ16は、ディスク記憶18-1-1および/またはメモリ記憶装18-1-2を含むキャッシュ記憶装置18-1の形式で、ドキュメント用の種々の記憶装置を有する。

クライアント12およびホームサーバ20では、従来技術と同様に、さまざまな情報（通常はドキュメントの形式）を分散するように動作する。このようなドキュメントは、実質的には、テキスト、グラフィックス、ピクチャ、オーディオ、ビデオ、コンピュータグラフィックス、および、コンピュータファイルもしくはコンピュータファイルの一部分に記憶される多くの種類の情報を含む。さらに、特定のドキュメントは、プログラムの実行により、アクセスがそのドキュメントを要求した時に作成されることもある。

次の説明においては、ネットワーク10はインターネットと仮定し、情報はハイパーテキスト転送プロトコル（HTTP）ドキュメントの形式にエンコードされ、ドキュメント要求メッセージは、TCP/IP層プロトコルを使用してユニフォームリソースロケータ（URL）の形式で送信される。これは、その他の種類のワイヤ式、スイッチ式、ワイヤレス式ネットワーク、およびFTP、ゴーパー（Gopher）、SMTP、NNTP等のようなその他の種類のプロトコルが、本発明を有利に使用できることを意味する。さらに、本発明はクライアント/サーバ形式の通信モデルのコンテキストで説明しているが、本発明の原理は、ピアツーピアネットワークにも同様に適合できることを理解すべきである。

特定のドキュメントに対する要求メッセージは、たとえば、クライアント12-1のようなクライアントコンピュータの1つで発生する。そのメッセージは、ホームサーバ20に対して、ディスクのようなホームサーバ20位置に現在記憶されているドキュメントのコピーを送れという、クライアント12による要求である。そのドキュメント要求メッセ

ージは、ルータ14-1、14-2、14-3のような1つ以上のルータ14を介して、図示した矢印の方向に転送されてホームサーバ20に達する。

インターネットのようなネットワークでは、ドキュメント要求メッセージは、ルータを介して、15またはそれ以上の数のノードまたは「ホップ」を通過した

後に意図した宛先に到達する。クライアント12-2、12-3、または12-4のような他のクライアントからの同ドキュメントに対する要求も、別のルータ14を通過して同時にホームサーバ20に到達する。

なお、ルータ14およびキャッシュサーバ16は、図1では別個の構成要素として示されているが、それらの機能は単一構成要素にまとめても良い。

あるモデルは、1つの特定のドキュメントに対する複数のクライアントからの要求が、コンピュータネットワーク10の経路をいかに経由するかを理解するのに役立つ。そのモデルは、メッセージがネットワークを通してホームサーバ20に伝送される際に、ドキュメント要求メッセージ上のルーチングアルゴリズムの効果から生じる構造Tである。図1に示すように、ホームサーバ20は、ルートから最も離れたリーフノードレベル、つまりクライアント12-1、12-2、...、12-4で発生したノードドキュメント要求と共に、構造Tのルートノードにあると考えることができる。構造Tは、また、ルータ14に配置された多くの中間ノードを有する。

クライアント要求が所定のホームサーバ20に向かって流れる経路セットの構造Tは、正確かつ一般的には、データダイレクト、非循環グラフとして示されるが、ここでの説明では複雑性を増すだけである。特に、特定の単一ドキュメントが1つだけのホームサーバに置かれていると考えられる場合、その構造は単一ルートを持つツリーと呼ばれる。このように理解して、ツリーの用語を、ここでは構造Tを説明するのに使

用する。このように理解すればグラフモデルも使用可能である。このモデルを念頭におくと、インターネット全体はツリーやグラフの森と考えられ、各ツリーやグラフは、ドキュメントの正式な複数組の永久コピーを提供する役割を持つ別のホームサーバ20に対する経路を持つ。

本発明によれば、ドキュメントのコピーはネットワーク中のキャッシュサーバ16に置かれている。本発明では、キャッシュコピーの配置（したがって負荷の拡散）は、ツリー構造T内のノードに制約される。これにより、クライアントは、ホームサーバ20もしくはドメイン名サービス（DNS）のようなネームサー

ビスと直接交信すること、またはネットワークを探索して適切なキャッシュコピーを探すことのいずれかによって、キャッシュコピーの位置を検索する必要がなくなる。

本発明は、また、ドキュメント要求メッセージがクライアント12からホームサーバ20まで自然に選ぶツリーに沿った経路上に、キャッシュサーバ16が存在すると仮定しており、キャッシュサーバは、ホームサーバ20の過大負荷を軽減したり、通信リンク自体のようなその他の潜在的な性能ボトルネックを拡散するのに協力する。実際には、キャッシュサーバに組み合わせたルータ14が、通過するドキュメント要求メッセージパケットを検査して、キャッシュされたドキュメントを提供することによって、実行できる要求をインタセプトする（横取りする）。

本発明の動作の最も一般的な説明では、ドキュメント要求メッセージは、クライアント12-3のようなメッセージを発生したクライアントからホームサーバ20に向かってツリーTをさかのぼって移動する。その経路に沿ってドキュメント要求メッセージを受けた特定のルータ、たとえばルータ14-7は、ローカルなキャッシュサーバ16を持たない。したがって、単にドキュメント要求メッセージを、ルータ14-6のようなツリー内の次のルータに渡すだけである。

しかし、ルータ14-6のようなその他の特定のルータは、ローカルキャッシュサーバを有しており、この場合、ドキュメント要求メッセー

ジは検査されて、ローカルキャッシュ記憶装置18に置かれたドキュメントを探るか否かが決定される。キャッシュコピーがキャッシュサーバで見つければ、そのコピーはクライアント12に返送され、要求メッセージは続けてホームサーバまでの経路上をたどることはない。しかし、キャッシュコピーが特定のキャッシュサーバ16-6で見つからない場合は、ドキュメント要求メッセージはホームサーバ20までの経路上における、次のルータ14-4に渡される。

ドキュメント要求メッセージパケットがルータ14に入ると、ルータは、まず、その要求メッセージを、ルータのソフトウェアの一部を構成する、本明細書ではフィルタコードと称する部分に渡す。ルータ14内のフィルタコードは、必

要に応じてローカルキャッシュサーバ16により更新される。フィルタコードは、パケットの種類、キャッシュ内容、ローカルキャッシュサーバ16の負荷、または接続されている通信リンク上の負荷に依存する。フィルタは、(ローカルキャッシュサーバにより提供されるサービスに対する)パケットをインタセプトするか、またはパケットをルータ14に送り返して、パケットがホームサーバ20までの経路を取る次のホップを決定する。

理想的には、キャッシュサーバ16の実装は、クライアント12またはサーバ20のいずれの通常動作モードに対しても、変更が要求されない。別の目的は、ネットワーク10の既存の下部構造に次第に用いられるように、設計をすることである。好ましくは、任意の新しいメカニズムが、既存の通信プロトコルと互換性を有する。

このためには、キャッシュサーバ16と組み合わせルータ14が、図2の階層図に示すように、4つの機能構成要素から構成されるのが好ましい。アプリケーション層のような相対的に上位層のプロトコルレベルにおいて、キャッシュサーバ16はHTTPプロキシ22とリソースマネージャ24とを含む。物理層のような下位層では、ルータは通常パケットフィルタ26とIPプロキシまたはスヌーパ28を実装する。

HTTPプロキシ22は、標準HTTPプロトコルを実装して、記憶管理とキャッシュされたドキュメントをアクセスするのに必要なインデックス構造の維持を含む役割を果たす。HTTPプロキシ22が、ローカルキャッシュ18に存在しないドキュメントに対する要求を受け取った場合は、ホームサーバ20にそのドキュメントを要求して、ドキュメントを受け取ると、要求に応答する。HTTPプロキシ22は、単にリソースマネージャ24により指示された通りにドキュメントをキャッシュするように構成されている。

図2では、2種類のプロキシ(代理)、つまりHTTPおよびIPレベルにおける代理を示すが、その実装は、アプリケーション層、IP層、またはトランスポート、セッション、プレゼンテーション等の別の層のプロキシを含むことができる。

リソースマネージャ24は、次に詳細を説明するように、ドキュメントコピーをネットワーク10を通して分散するプロトコルを実装する。リソースマネージャ24は、ドキュメント負荷分散メカニズムで使用する状態情報を保持する役割を持つ。リソースマネージャ24は、キャッシュサーバ16自体の負荷を管理するだけでなく、ルータ14を相互接続するために使用する通信経路上のトラフィックを管理するようにプログラムされている。

この負荷管理、つまり負荷バランスを達成するために、リソースマネージャ24は、近接するキャッシュサーバ30のアイデンティティ（実体）および負荷に関する情報を保持する。どのように近接のキャッシュサーバ情報を保持するかの詳細については、次に第3節で説明する。

さらに、キャッシュ18内の各ドキュメントに対して、リソースマネージャ24は、近接のキャッシュサーバ30の各々から受け取る要求を区別する。これは、各近接キャッシュサーバ30から受け取る要求に対するヒット数を、別々に保持することでなされる。このような情報を用いて、リソースマネージャ24は、分散する過大負荷部分を計算する。

これらの負荷部分を求めると、リソースマネージャ24は、キャッシュすべきドキュメントと受け取るべき要求の負荷部分を近接の軽負荷サーバ30に通知する。この負荷部分は、組み合わせたルータ14に導入される新しいフィルタコードを生成するのにも使用される。同様の方法が、軽負荷の近接サーバ30でも実行される。必要なら、軽負荷のホームサーバ20のリソースマネージャ24は、接続されているHTTPプロキシ22に、キャッシュ18に新しいドキュメントを追加するように通知する。

リソースマネージャ24のその他の役割は、近接サーバの発見、近接サーバ30に対して負荷情報の普及、潜在的障害の発見と負荷平均化への回復である。これらのメカニズムは次にさらに詳しく説明する。

ルータ14は、キャッシュサーバおよび／または通信経路の平均化の目的を遂行するキャッシュサーバ16を支援するのに積極的な役目を果たす。これは、リソースマネージャ24が、フィルタ26およびスヌーパ28を実装するコード形

式で機能をルータ14に導入させることで、遂行される。特に、ホストサーバ20に対して直接アドレス指定されておらず、ルータ14を通過する全パケットは、先ずスヌーパ28に渡される。スヌーパ28は、パケットを検査して、その種類、宛先、要求されるドキュメントを判定する。キャッシュサーバ16の状態とパケットタイプに応じて、スヌーパ28は、パケットをインタセプトするか、またはホームサーバ20までの目的の宛先パスに沿って、次のホップまたはルータ14にパケットを単に送出する。

要求されたドキュメントがローカルキャッシュサーバ16に存在するかを判断するために、スヌーパ28はフィルタ26に問い合わせる。フィルタ26が、要求されたドキュメントがキャッシュされてローカルにサービスできることを示すと、パケットはインタセプトされて、リソースマネージャ24に渡される。そうでない場合は、パケットは宛先のホームサーバに向かって、次のホップに渡される。

スヌーパ28は、通常、TCPプロトコルとTCPおよびHTTPパケットの両方の構造とを認知している。スヌーパ28の別の機能として、HTTP要求パケットを抜き出して、リソースマネージャに渡すことがある。この機能は、リソースマネージャ24が、近接サーバを検出し、潜在的な障害から回復することを支援するために使用される。

## 2. TCP/IPネットワーク内のHTTPドキュメント要求の処理

HTTPを使用するネットワーク10の現在の実装は、クライアントとサーバ間の高信頼性の端末間通信用の階層化されたTCP/IPプロトコルに依存している。この階層化は、要求メッセージの通常処理を3段階に分割している。つまり、コネクションの確立（すなわち{SYN}メッセージ形式のTCPレベルのスリー・ウェイ・ハンドシェイク）、{GET}メッセージ形式のHTTPドキュメント要求/応答、および{FIN}メッセージの形式のコネクションの終了である。

この処理は図3に示すとおりである。クライアント12は、まず、ホームサーバ20に、シーケンス番号を含む{SYN}メッセージを発行し、ホームサーバ

20は、確認応答{ACK}と共に{SYN}メッセージを返す。これに応答して、クライアント12は、所望するドキュメントのURLを含む{GET}メッセージの形式のドキュメント要求を送信する。ドキュメントは、ホームサーバ20によって、クライアントに送出される。クライアント12が確認応答を返した後に、サーバ20とクライアント12は、{FIN}と{ACK}メッセージを交換して、コネクションを終了する。

上述したように、このような環境でキャッシュサーバ16を実際に実装させる上での一番の障害は、クライアント12から要求されたドキュメントを、キャッシュサーバ16が識別しなければならない、という要求である。しかし、図3に示すように、URL情報は、通常、TCP/IPコネクションがクライアント12とホームサーバとの間で確立し

た後に、HTTPクライアント12によって伝えられる。1つの解決法は、このようなコネクションすべてをホームサーバ20との間に確立させて、中間ルータ14のスヌーパ28にすべての{GET}パケットをインタセプトさせることである。この方法は、ホームサーバから大量の負荷を取り除くが、ドキュメントに関係するTCPコネクションをホームサーバにまで到達させるので、ホームサーバ20の負荷を軽減させる本来の目的を失わせてしまう。需要が集中する間は、多数の要求によって、ホームサーバ10に{SYN}要求が殺到する。さらに、最初の{SYN}がインタセプトされない場合は、コネクションの確立と解除の両方がさらに著しく複雑になる。

この障害を克服するために、好ましい実施形態では、中間ルータ14がTCPプロトコルをある程度認識している。TCP認識ルータ14は、すべてのHTTPサーバに対するTCPコネクション要求(つまり、HTTPポートに向かう{SYN}パケット)を検出でき、ホームサーバ20のプロキシ(つまり「擬似」として動作することができる。

この機能はスヌーパ28により実装されている。特に、スヌーパ28はHTTPホームサーバ20への経路上のルータ14内に配置されて、ルータ14を通過していくパケットを検査してパケットを識別する。さらに、HTTPホームサーバ



20に向かう任意の{SYN}パケットをインタセプトする。{SYN}パケットはクライアント12が要求しようとしているドキュメントの情報を含まないので、スヌーパ28はホームサーバのプロキシ、つまり「擬似」として動作する。この動作には、クライアント12とキャッシュサーバ16のローカルトランスポート層間のコネクションを確立し、クライアント12とローカルトランスポート層の両方で使用する初期シーケンス番号を控える。

コネクションの確立後、スヌーパ28は通っていくすべてのパケットを検査し、対応する{GET}要求を待つ。{GET}要求が到達すると、スヌーパ28はローカルフィルタ26とリソースマネージャ24に

、要求されたドキュメントがキャッシュされているかを問い合わせる。ドキュメントがキャッシュされていれば、スヌーパ28はHTTPの{GET}メッセージをローカルリソースマネージャ24に送出し、リソースマネージャ24が要求をサービスするのを待つからコネクションを終了する。そうでない場合は、要求されたドキュメントはキャッシュされていない（たとえば、フィルタ26またはリソースマネージャ24がミスする（見つけられなかった））。この時点で、ドキュメント要求にサービスするために、いくつかの異なる方法を採用できる。

第1の方法では、TCPコネクションをはずす。すなわち、スヌーパ28は、クライアント12との擬似TCPコネクションのサーバ部分を終了し、合成「ピギーバック」{SYN+GET}メッセージ（つまり、{SYN}メッセージに{GET}メッセージが埋め込まれたもの）形式でホームサーバ20に向かうドキュメント要求を送出する。さらに、{SYN+GET}メッセージは、要求されたドキュメントをキャッシュするホームサーバ20への経路上の、任意の別の中間キャッシュサーバとのTCPコネクションのサーバ部分をはずすのに必要なすべての状態情報を含む。

第2の別の方法は、スヌーパがTCPリレーとして動作するもので、クライアントとのTCPコネクションは維持し、分離したコネクション上で、{SYN+GET}メッセージを、ホームサーバ20への経路上の次の中間キャッシュサーバに中継する。

上述のTCPコネクションをはずす方法を図4にフローチャートで示す。この方法は、リーフノードサーバ38と呼ばれる、特定のクラスのキャッシュサーバ16で実行され、このリーフノードサーバ38は、ツリーTの最下位レベルノードに存在する、つまりクライアント12からの{SYN}パケットを最初にインタセプトするキャッシュサーバ16である。図1において、ツリーT内のリーフノードサーバ38は、キャッシュサーバ16-1、16-6、および16-8である。

図4のステップ41に示すように、リーフノードサーバ38が{SYN}パケットを受信すると、ホームサーバ20はTCPコネクションをリーフノードサーバ38とクライアント12間に直接確立して、ホームサーバの代理をする、つまりホームサーバと見せかける。その後、リーフノードサーバ38は、クライアント12からの応答{GET}要求をインタセプトするため待機する。

なお、クライアント12とキャッシュサーバ16間で交換されるパケットが、上述のようにスヌーパ28によって修正されて、このように擬似（見せかけること）が起こる。特に、要求をサービスするキャッシュサーバ16のネットワークアドレスは、ホームサーバ20のネットワークアドレスと置き代わり、コネクションははずされている。また、キャッシュサーバ16が発行するシーケンス番号バイトは、リーフサーバ38が決定するシーケンス番号に続く。

ステップ41に戻って、要求されたドキュメントがフィルタ28によるキャッシュ問い合わせテストに合格し、ステップ42に入る。リソースマネージャ22が、要求ドキュメントがローカルキャッシュ内に存在することを検出し、そのドキュメントへのアクセスを許可すると、ドキュメント要求はローカルにサービスされる。つまり、リーフサーバ38で処理される。ステップ45では、{GET}コマンドはリソースマネージャに送出され、要求ドキュメントに応答する。最後に、再度ホームサーバ20と見せかけて、終了の{FIN}と{ACK}メッセージをクライアントに発行することで、リーフサーバ38とクライアント12間のTCPコネクションを終了する。

一方、ステップ42と43でミスがあると、スヌーパ28は{SYN+GET}

｝パケットをホームサーバ20に向けて送出し、そのサーバを擬似TCPコネクションのサーバ部分を終了する。これにより、ツリー上の別のキャッシュサーバが、可能であれば、サービスする。図4のステップd)とe)は非同期なイベントであって、通常は同時に起こる。

次に、リーフサーバ38のスヌーパ28は、{GET}要求の受信を確認する必要がある。

図1のネットワークでは、上流側中間ノンリーフノード39は、キャッシュサーバ16-3、16-4、および16-10を含む。ノンリーフノード39のキャッシュサーバ16は、少し異なった方法で、{SYN+GET}パケットの処理を必要とする。特に、ノンリーフノード39内のスヌーパ28は、要求されたドキュメントがキャッシュされて、ローカルキャッシュサーバ16がサービスするのに十分な容量を持っている場合のみ、{SYN+GET}パケットをインタセプトする。

図5は、ノンリーフ中間ノード39で実行される処理の詳細なフローチャートである。ステップ51に示すように、要求に対するサービスをするために、スヌーパ28は先ずリーフノード38から{SYN}を代理で受け取り、次の{GET}要求をインタセプトする。次のステップ52および53では、リーフノード38と同様に、フィルタ26とリソースマネージャ24に問い合わせして、{GET}がローカルに処理できるかどうか決定する。

その要求がローカルに処理できる場合は、ステップ55では、クライアント12間のTCPコネクションのサーバ部分を確立し、{GET}をリソースマネージャ24に発行し、ドキュメントをクライアント12に返送し、TCPコネクションを終了することによって、ホームサーバ20の代理を完了する。

{GET}メッセージがローカルに処理できない場合は、ステップ54が実行され、{SYN+GET}がツリーTの次のノードに送出される。

中間ノンリーフノード39内で異なった方法で{SYN+GET}パケットを処理する主な利点は、実際に要求されたドキュメントを有する特定の中間ノード39とのTCPコネクションが一度はずされるだけである。別の利点は、{SY

N+GET} が、コネクションをはずすのに

必要なすべての状態情報を含むことである（つまり、リーフノードサーバ38のスヌーパ28と、要求されたドキュメントを実際にキャッシュする中間ノード39のスヌーパとの間で、さらに状態情報の交換をしない）。

この方法におけるピギーバック {SYN+GET} パケットの欠点は、このようなパケットを処理するトランスポートプロトコルを採用しなければ、ホームサーバ20がこのようなパケットを適切にインタセプトしないことである。この問題を避けて現在のネットワークプロトコルでの相互操作性を確実にするには、追加対策が必要である。それためには、スヌーパ28をホームサーバ20の前の最終中間ノード39に置き、すべての {SYN+GET} パケットをインタセプトする。したがって、リーフノードサーバ38または中間ノードサーバ39のいずれにも要求されたドキュメントがキャッシュされていない場合、最終中間サーバ39が {SYN+GET} をインタセプトして、明白なHTTP {GET} 要求をホームサーバ20に中継する。

このような場合に適合させるために、図5のステップ54を図6の方法に置き換える。その場合、ステップ61において、経路Tに沿った次の上流ノード（または親ノード）がホームサーバ20でなければ、ステップ62に入り、{SYN+GET} がT上の次の中間ノードに送出される。

しかし、次のノードがホームサーバ20の場合は、ステップ63が実行される。特に、スヌーパ28はクライアント12とTCPコネクションのサーバ部分を確立し、ローカルリソースマネージャ24に対する {SYN+GET} を {PROXY-GET} 要求に置きかえる。リソースマネージャ24は、{PROXY-GET} 要求をホームサーバ20に向けて発行する明白な {GET} に変換する。その後、ホームサーバ20の応答はクライアント12に中継され、キャッシュサーバが要求されたドキュメントをキャッシュしていたかのようにする。

これまで説明したキャッシュ技術の別の欠点は、特定のクライアント12とホームサーバ20間のツリーTに沿った経路が、リーフノードサーバ38または中

間ノードサーバ39が要求へのサービスを決定した後に、代わりうることである。これは、たとえば、ネットワーク接続またはリンクが2つのサーバノード間で失われる時に起こる。図7は、この比較的まれな事例を示している。ここで、クライアント12とホームサーバ20間の経路は、中間キャッシュサーバ16bがクライアント12からのドキュメント要求を処理している間に代わってしまう。その結果、クライアント12から送信されるすべての{ACK}は、新しい経路上を移動し、新しいキャッシュサーバ16xを通してホームサーバ20に到達する。これにより、キャッシュサーバ16bはタイムアウトして、そのパケットを再送信してしまう。

この問題を解決するために、サーバ16bのスヌーパ28が、パケットが再送信された回数を記録する。パケットが所定の回数、例えば3回よりも多く再送信された場合、スヌーパ28は、クライアント12とホームサーバ20間の経路が代わったと見なし、クライアント12とのコネクションを終了するステップを取る。特に、スヌーパ28はクライアント12とのコネクションを中止してキャッシュサーバ16bとのコネクションを中止し、同時にホームサーバ20と見せかけて、リセットパケット（例えば、{RST}パケット）をクライアント12に送信する。

別の方法では、クライアント12に最も近いリーフノードサーバ38とサーバ20に最も近い最終のホップノードには、クライアント12とサーバ20それぞれに対して、一つのルートのみが設けられている。これは、単に要求メッセージを通常のルートで送出させる代わりに、キャッシュサーバにクライアントの要求メッセージをキャッシュサーバ-キャッシュサーバ間の固定したTCPコネクションを介して送出させて行う。したがって、一連の適切に結合されたTCPコネクションとして実

装されているコネクションセットは、ネットワーク構成が変化した時、IPルーティング（IP経路選択）の変化に自動的に適応する。

### 3. 近接発見（近接サーバの発見）

しかし、近接キャッシュサーバとの通信により、近接キャッシュサーバの構成

のすべての変化を検出し、資源負荷の平均化と本発明により可能なその他の利点が達成される。特に、上述の方法に関係する各キャッシュサーバ16は、他のどのサーバが近接しているかを判定する。さらに、各ルーチングツリーT上で、キャッシュサーバ16は上流側サーバ（親ノードに設置）と下流側サーバ（子ノードに設置）を区別する。ツリーT内の特定のノードiはノードjの親ノードである。jからホームサーバ20方向のルート上で、iが最初のキャッシュサーバ16ならば、この場合はノードjはノードiの子と呼ばれる。

キャッシュサーバ16がその近接サーバを発見する1つの方法は、下位ルータ14とスヌーパ28の支援を必要とする。選択した時間に、リソースマネージャ24はローカルルータ14に、そのルータが保持しているルーチングテーブル内の各宛先に対して近接発見メッセージを発行するように求める。

次に、これらの近接発見パケットは、ツリー内にキャッシュサーバ16を有する別のノードに存在する所定のスヌーパによりインタセプトされる。近接発見パケットを発行するキャッシュサーバ16のリソースマネージャ24に応答を送信するのは、インタセプトするキャッシュサーバ16の役割である。応答において、親であること（例えば、そのパケットを発行しているキャッシュサーバよりもホームサーバ20により近いこと）、およびツリーTのアイデンティティ（実体）を知らせる。近接発見パケットに対する宛先ポートは、存在しえないポート番号を指定してもよい。これは、宛先ホームサーバ20が、インタセプトされていない近接パケットを、間違って処理しないようにするためである。また、ホップカウントフィールドは、近接発見パケットを過剰に送出される

のを制限するのに、つまり離れたノードまで送出されるのを防止するのに使用される。

この方法の欠点は、近接発見パケットがネットワークに溢れる可能性があることである。別の方法は、各キャッシュサーバ16内のフィルタを通過するドキュメント要求メッセージパケット（すなわち、{SYN+GET}パケット）を使用する。

この方法では、各要求メッセージは前のホップを識別するフィールドを含み、

上述の実行した方法で、特定の要求パケットが通過した最終サーバ16の識別が行われる。

要求がルーター12を通過する時（つまり、インタセプトされない）、ローカルスヌーパ28は、付帯しているキャッシュサーバ16のIPアドレスをスタンプする。キャッシュサーバ16がその近接サーバを発見しようとするとき、キャッシュサーバ16は、付帯しているスヌーパ28に、要求パケットから最終監視宛先と最終ホップアドレスとを抜き出すように指示し、その後この情報をローカルリソースマネージャ24に渡す。

図8に示すように、代表的なHTTP {GET} メッセージがクライアント12からAを通ってホームサーバ20に経路を流れ、中間キャッシュ16cによりインタセプトされる。キャッシュサーバ16cが要求を処理している間、ホームサーバ20とクライアント12間の経路は、すべての確認応答に対して、別の経路を使用させる。

この情報を使用して、キャッシュサーバ16cのリソースマネージャ24は、キャッシュサーバ16が存在しているルーチングツリーと、ツリー上のすべての下流側キャッシュサーバ16との両方を求める。サーバ16cが、サーバ16bがツリーT上の下流側の子と判定すると、キャッシュサーバ16cは、自身がT上の親であることをキャッシュサーバ16bに明確に通知する。異なる構成要素間（スヌーパ28とリソースマネージャ24）のメッセージ交換回数を減らすために、スヌーパ2

8は多数のパケットをキャッシュして、それらすべてを1回でリソースマネージャ24に送出できるようになっている。

近接情報は、近接発見の所定の世代数、たとえば世代だけ保持される。この期間中に、どんな要求も子キャッシュサーバ16bを介して受信しない場合は、子キャッシュサーバ16bを近接のキャッシュサーバ16cのモデルから除外する。次に、親キャッシュサーバ16cは、子キャッシュサーバ16bに、モデルから除外する意図を通知する。

キャッシュサーバ16が、ホームサーバ20へのルーチングツリー上に、親ス

スヌーパ28を持たないこともある。この場合、キャッシュサーバ16bのスヌーパ28は、近接発見 packets をホームサーバ20に向けて送出する。サーバ16cのような上流側スヌーパは packets を受信し、16bに自身がホームサーバ20に対するツリー上の親であることを通知する。しかし、キャッシュサーバ16bのスヌーパ28が、ホームサーバ20に対するツリー上にキャッシュサーバ16cのような親ノードを持たない場合は、スヌーパ28が近接発見 packets 上の16bアドレスを置換して、それをホームサーバ20方向に送出する。

この近接発見方法は多くの利点を持つ。第1は、ルーチングツリーTが、動作を開始するためのキャッシュプロトコルに対して完全に構成されていなくてよいことである。別の利点は、協働するキャッシュサーバ16は動的な発見ができ、ルート変更に適応できることである。最終的にはプロトコルが全体に分散され、これにより、サーバ故障に対して強くなる。

#### 4. 負荷の平均化 (ロードバランス)

大部分の他のキャッシュ方法と異なり、本発明によるキャッシュ方法は、実際にクライアントからの要求がある前に、キャッシュサーバ16へのキャッシュコピーの分散を必要とする。すなわち、クライアント12からの特定のドキュメント要求のすべてに直接応答するよりは、将来

のドキュメント要求を予測してドキュメントをキャッシュサーバ16間で移動させる。

上述のドキュメントキャッシュと近接発見の方法は、キャッシュ負荷の分散および/またはキャッシュサーバ16およびそれらを相互接続する通信経路の両方の負荷平均化のような、多数の異なる種類の目的に適合する。好ましい実施形態において、この負荷分散方法では、厳密にはローカル情報に依存する、拡散ベースのキャッシュアルゴリズムを使用して、キャッシュシステムのサイズを急激に拡大するオーバーヘッドの発生を避ける。

特に、ルーチングツリーTの上流側(親)ノードが、軽負荷の下流側(子)ノードまたはリンク(これに対しては、リソースマネージャが、キャッシュされたドキュメントの1つを与えてドキュメントサービス負荷のいくつかを移動できる



)を検出したときだけ、リソースマネージャ24はキャッシュコピーを作成する。ここでは負荷は、要求実行率、クライアント応答時間、サーバ動作時間の一部のような多数の異なる性能の基準となる。

逆方向における負荷の不均一は、子ノードから、キャッシュされたドキュメントのいくつかを消去させたり、サービスしようとするこれらドキュメントに対する要求の一部を減少させたりする。

通常、一定の判断基準に従って子ノードから頻繁に親ノードに要求されるドキュメントは、軽負荷の子ノードに向けられる。

同様に、キャッシュサーバ16がローカルキャッシュ18からドキュメントをなくす必要があるのは、そのドキュメントに通常ほとんど要求がないときである。

さらに、負荷を近接サーバに移管するときは、キャッシュサーバ16はドキュメントを子または親にそれぞれ押しつけ、または開放(放出)できる。拡散メカニズムによるこのドキュメントキャッシュの目的は2つある。それらは一般に従来の拡散方法にはないものである。これは、

キャッシュサーバが複写するドキュメントを決定する他に、ドキュメント要求のどの部分(つまり、負荷)を軽負荷の近接16に移管すべきか決定する必要性のなかで明白にされる。

第1の目的は、キャッシュサーバが軽負荷の近接サーバにドキュメントを渡す方法を決定することである。ここでの目的は、ネットワーク内のキャッシュサーバ16すべての最大容量を引き出すことである。第2の目的は、人気のあるドキュメントをクライアント近くに移動し、人気のないドキュメントをクライアントから離し、複写の作成を最小にして、応答時間を短縮することである。これらの目的は、キャッシュサーバ16間の通信経路負荷をも考慮して、達成される。

これらの目的を実行するには、ノード*i*の過負荷のキャッシュサーバが、次式のように、子ノード*j*の最小軽負荷キャッシュサーバを判定する。

$$L_j \leq L_k, \forall k \in C_i$$

ここで、 $L_i$ は特定のキャッシュサーバー  $i$  の負荷、 $C_i$ は全キャッシュサーバーの集合である。過負荷キャッシュサーバー  $i$  は、子  $j$  に最も人気のあるドキュメントに対する要求の一部を押しつける。特に、次の特性を持つ場合、ドキュメント  $d$  はノード  $i$  からノード  $j$  に押しつけられる。

$$P_{ji}(d) = \max_{g \in D_i} P_{ji}(g)$$

ここで、 $D_i$ はノード  $i$  にキャッシュされているドキュメントの集合である。一方、次式を満たす場合は、ドキュメント  $d$  はノード  $i$  からノード  $j$  に開放（放出）される。

$$P_{ji}(d) = \min_{g \in D_i} P_{ji}(g)$$

この方法は最大-最小ドキュメント拡散と呼ばれ、上で述べた最初の2つの目的を満足する。

所定のキャッシュは、大部分が負荷軽減に寄与するドキュメントを一杯にする、ビンと見なすことができる。これにより、所定のドキュメントに関する負荷が増すにつれ、特定のキャッシュ内に置く必要のあるドキュメントの数は小さくなる。最大-最小ドキュメント拡散は、この第3の条件を満たすために、最小のドキュメント数を提供する。

拡散するドキュメントを決定すると、キャッシュサーバは、近接サーバにその負荷を移管するときにインタセプトする要求の一部を変更する方法を決定する必要がある。この部分は、キャッシュサーバがドキュメントによる負荷の一部を移管する方法を考慮して決定できる。特に、ノード  $i$  がノード  $j$  の親で、 $i$  が  $j$  に比べて過負荷であると仮定する。さらに、ノード  $i$  がドキュメント  $d$  をノード  $j$  に提供する要求の全数の  $R_{ij}(d)$  に等しい複数の要求を押しつけたいと要望していると仮定する。要求の全数を次のように定義すると、

$$\beta_{ji}(d) \Delta_{ji}(d)$$

ドキュメント  $d$  に対してノード  $i$  がサービスする要求数は次式で与えられる。

$$P_{ij}(d) = \sum_{m \in C_i} \beta_{mi}(d) \Delta_{mi}(d)$$

$R_{ij}(d)$  要求をノード  $j$  に押しつけた後に、ノード  $i$  は次式で与えられる複数の要求をサービスする。

$$P'_{ii}(d) = \sum_{m \in C_i} \beta_{mi}(d) \Delta_{mi}(d) - \beta_{ji}(d) \Delta_{ji}(d) + \beta'_{mi}(d) \Delta'_{mi}(d)$$

ここで、

$$\Delta'_{ji}(d) = \Delta_{ji}(d) - R_{ij}(d)$$

$\Delta_{ji}(d)$  は、ノード  $i$  への移管が完了後に、ノード  $j$  でインタセプトされない要求の数を表す。 $\Delta'_{ji}(d)$  については、ノード  $i$  でインタセプトされた新しい部分が  $\beta'_{ji}(d)$  で表される。次式が成立する。

$$R_{ij}(d) = \beta_{ji}(d) \Delta_{ji}(d) - \beta'_{ji}(d) \Delta'_{ji}(d)$$

上の値  $\Delta'_{ji}(d)$  と  $R'_{ji}(d)$  を用いると、簡単に次のようになる。

$$R_{ij}(d) = \beta_{ji}(d) \Delta_{ji}(d) - \beta'_{ji}(d) (\Delta_{ji}(d) - R_{ij}(d))$$

代数的計算により、次の新しい値となる。

$$\beta'_{ji}(d) = \frac{\beta_{ji}(d) \Delta_{ji}(d) - R_{ij}(d)}{\Delta_{ji}(d) - R_{ij}(d)}$$

新しい部分  $\beta'_{ji}(d)$  を計算後、ノード  $i$  は組み合わされたフィルタを更新し、ドキュメント  $d$  のコピーをノード  $j$  に送出し、ノード  $j$  にノード  $j$  が  $R_{ij}(d)$  によってインタセプトされる要求の数を増加するように指定する。

ノード  $j$  がこの情報を受け取ると、次式を計算し、

$$\beta'_{mj}(d) = \frac{R_{ij}(d)}{\Delta_{ji}(d)} (1 - \beta_{mj}(d)) + \beta_{mj}(d), \forall m \in C_j$$

これらの更新内容をそのローカルフィルタコードに反映させる。なお、解析を

完成するには、ノード  $j$  でフィルタされる要求の数が  $R_{ij}(d)$  で増加する。次式でも知られている。

$$P'_{ij}(d) = \sum_{m \in C_j} \beta'_{mj}(d) \Delta_{mj}(d)$$

$\beta'_{mj}(d)$  の新しい値を代入すると、次式が得られる。

$$P'_{ij}(d) = \sum_{m \in C_j} \left[ \frac{R_{ij}(d)}{\Delta_{ji}(d)} (1 - \beta_{mj}(d)) + \beta_{mj}(d) \right] \Delta_{mj}(d)$$

代数的計算により、

$$\begin{aligned} P'_{ij}(d) &= \frac{R_{ij}(d)}{\Delta_{ji}(d)} \Delta_{ji}(d) + P_{ij}(d) \\ &= P_{ij}(d) + R_{ij}(d) \end{aligned}$$

このようにして解析を完了する。

大きなドキュメントでは、ローカルキャッシュ 18 が要求されるドキュメントの一部だけを含むのが有利である。この場合は、キャッシュサーバはドキュメントのローカルにキャッシュされた部分をクライアント 12 に提供することを開始でき、一方同時にドキュメントの残り部分をホームサーバから送るように要求できる。

その他の基準は、ドキュメントのサイズ、ドキュメントの種類、伝送方向、または特定のドキュメントに関連する優先度である。

キャッシュサーバ 16 とルータ 14 は、通信バッファを使用して、高い優先度のメッセージのトラフィックを向上させることもできる。

同一種類の関係するドキュメント、またはキャッシュサーバ 16 により、いくつかの点で通常密に連続して要求されるのが認められた関係するドキュメントは、同一動作でキャッシュサーバ間を移動する。

このように各キャッシュサーバ 16 には、そのローカルな負荷、近接

負荷、通信経路負荷、ならびに人気度合い、サイズ、フェッチコスト等のドキュメント属性に基づいて、ドキュメントのキャッシュと放棄の機能が与えられる。特に各サーバは、その近接サーバの負荷の評価結果を維持し、および／またはさまざまな時間にその情報または実際の負荷の情報を近接キャッシュサーバ16に転送、または「ゴシップ」する。キャッシュサーバ16が、近接サーバに比較してどれかの点で自身が過負荷であると認識すると、将来予測される仕事の一部分を、上に述べたように、軽負荷の子または親の近接サーバに移管する。

このように本発明は負荷分割に適し、所定のサーバに対するルーチングツリーT内の同一パス上の近接（「同胞」）ノードは、特定ドキュメントに対する要求負荷を分担する。

上記のドキュメント拡散アルゴリズムは、最適な負荷分散にある種の難題を伴う可能性がある。特に、所定のキャッシュサーバjが潜在的な障害になるのは、最低限2つの子キャッシュサーバkおよびk'、ならびに親キャッシュサーバiを持ち、その各サーバの負荷Lが次式を満たすときで、

$$L_k' \leq L_j \leq L_i \leq L_k$$

キャッシュサーバjは軽負荷の子kが要求するどんなファイルもキャッシュしない。

図9(a)は、そのような場合の例を示す。キャッシュシステムは、(ノード番号1において)ホームサーバ20と、3台の中間サーバ39(ノード2、3、4とする)から構成される。要求はリーフノードのみで発生し、特にドキュメントd\_\_1とd\_\_2はノード4から要求されており、d\_\_3はノード3から要求されている。図は特定ノードのキャッシュコピーの配置と各キャッシュされたコピーによりサービスされる要求

を示す。

この例では、ノード2のキャッシュサーバ16は潜在的な障害になる。このサーバは、d\_\_3をキャッシュしないので、その負荷をノード3のキャッシュサーバに拡散できない。さらに、ノード2のキャッシュサーバ16は、ノード1のキ

キャッシュサーバが問題の存在を認識することを妨げている。

1つの可能な最適負荷配分は、負荷を各ノードサービス要求に合わせて4つのノードに等しく分散することである。図9(b)は、この条件を満たすファイルキャッシュと負荷分散を示す。

しかし、拡散を基本とした方法は、キャッシュサーバ16が図9(a)のような不都合な状態を検出してそれを回復できるように、変更することが可能である。特に、jが軽負荷を修正するアクションを取ることなく、kがjに比べて所定の時間回数(たとえば2)以上に軽負荷に留まるときは、キャッシュサーバkはその親サーバjが潜在的に障害であると仮定できる。図9(a)の例では、キャッシュサーバkはノード3に対応し、キャッシュサーバjはノード2に対応する。親ノードkから拡散された負荷が無いことを検出すると、子ノードは親ノードがkにルートをもつサブツリーから要求されたドキュメントをキャッシュしていないと推測する。これらドキュメントのコピーが最終的に提供されると、サーバkは通常通りそれをキャッシュできる。この場合サーバkは負荷障害となる親ノードjが存在する状態でも、コピーを取り込んでキャッシュできるため、この技術はトンネリングと呼ばれている。

## 5. その他の機能

本発明はネットワークで追加機能を実装するのに使用できる。これらの各種機能は、ルータ14に設置されているフィルタ26が、ローカルキャッシュサーバに対するドキュメントのコピーの要求を単に転送する以上の役割をこなすという事実による。

たとえば、どのドキュメントをどのサーバにキャッシュするかを選択を制御するためには、ドキュメントの人気統計がキャッシュサーバ16で収集される必要がある。各ノードのキャッシュサーバ16は特定のドキュメントに対する参照回数とそれがどこから来たのかを記録し、要求総量を集計する。

このデータは中央位置に収集され、特定のネットワーク領域に設置されたキャッシュサーバ16のグループによるのと同様に、キャッシュサーバ16のグループによって整理される。このように収集したデータにより、特定のクライアント

でその資料に対する「ヒット率」の統計を得たドキュメントだけでなく、ヒットが発生したネットワーク10の下位部分の発行も可能である。

また本発明では、需要を予測して、ネットワーク10中に一種のドキュメントの加速分散ができる。

キャッシュサーバ16は、サービスプロバイダの検知から負荷分割装置としてデータベース、索引ファイルの検索、その他の人気のあるドキュメントのホストの複製に使用できる。すなわち、データベースプロバイダは、ネットワーク10内にドキュメントを配置して持つことができ、配置がどのようなであろうと、アクセスを希望するクライアントにより近いデータを送出できる。

また、キャッシュサーバは、要求メッセージとその他の情報の発生源の確認を実行できる。これはキャッシュサーバ16が、特定のドキュメント要求メッセージの発生源のクライアントについては自動的に情報を保持することでなされる。クライアント12がそのドキュメントに対し許可された要求者でない場合は、その要求メッセージは、ホームサーバ20に到達する前に、キャッシュサーバ16で終了させることができる。

選択セキュリティも設けられる。そのメカニズムはノードの各々がフィルタ26、リソースマネージャ24、キャッシュ収納装置18および

HTTPプロキシ22を備えているからである。フィルタ26は、キャッシュコピーに対する要求をHTTPプロキシ22にリダイレクトする方法を記録するだけでなく、その情報に対する要求を認証することで、キャッシュコピーへのアクセスするのに使用される。キャッシュサーバ16間のリンクが容易に高信頼性リンクとして構成できるため、クライアント12からサーバ20への経路に沿った最初と最後のホップは信頼性が高い。安全確実なリンクがクライアント12とホームサーバ20間をキャッシュサーバ16を経由して提供される。

さらに一般的には、キャッシュサーバ16は、ホームサーバ20による分散とその後のクライアントへの配布の間、いくつかの時点でドキュメントを変換する。このような変換は、ドキュメントまたはプログラムをキャッシュするために値に加えるいくつかの機能を可能および／または実装でき、ドキュメントまたは／

およびプログラムの圧縮や暗号化のような機能を含む。さらに、既知のセキュリティ技術が発生源、宛先または両方（取引業務の場合）において、変換として適用できる。暗号化と解読の他に、これらは認証、許可（アクセスコントロール）、拒絶禁止、保全性管理を含む。これらのセキュリティ技術は、通常、鍵を要求する暗号化技術を使用でき、さらに、キャッシュサーバへの物理パスに使用して、認証、許可、拒絶禁止を強化できる。

通常の用途では、これらの変換は、それぞれに整合を取られ（例えば、各暗号については、一致解読であるべき）、個別に使用される。しかし、それらはドキュメントの寿命の点では異なって構成されている。たとえば、ドキュメントは、すべて異なる時点で、圧縮、暗号化、解読、および解凍される。

図10は、ドキュメントを変換するのに適用する1つの可能な方法を示す。この例では、ドキュメントがホームサーバ20から配布され、クライアントに送られる時に、明確な変換が別個に4回生じている。つまり、

T1において、要求がクライアント12で作成されるとき

T2において、要求がサーバ20に送出されるとき

T3において、応答がサーバ20により送られたとき

T4において、応答がクライアント12に送出されたとき

図10の変換が通信の4つのフェーズに関連しているとしても、実際には、変換はノード自体により実行される。このように、要求受信後のキャッシュサーバ16-6によって、または要求送信前のクライアント12によって、T1はその要求で実行される。前者の場合、キャッシュサーバ16-6が変換を実行している。後者の場合は、クライアント12が変換を実行する。まれにキャッシュサーバ16-6を通してこれをトンネリングする。同様に、要求がホームサーバ20に送出される際に、変換T2がキャッシュサーバ16-7またはホームサーバ20で実行される。応答がホームサーバ20から送信され、最終的にはクライアント12に送出される際に、同様であって逆のことがT3とT4に適用される。

図10で実行される変換は、セキュリティ用（またはドキュメント以外の入力を要求する別の変換）の鍵の配置において、重要な役割を持つ。キャッシュサーバ



バ16自体がセキュリティを実装する場合は、暗号化鍵は、必要に応じてキャッシュサーバ16に分散されなければならない。この場合、端末間のセキュリティは、クライアント12とキャッシュサーバ16-6間のほかに、キャッシュサーバ16-7とホームサーバ20間にもセキュリティチャネルを追加して提供される。クライアント12とホームサーバ20が、それら自体で端末間セキュリティを実装する場合は、キャッシュサーバ16は妨害せずに、鍵をクライアント12とホームサーバ20に配布するのに可能な限り協力する。

図10は、ネットワーク端にキャッシュサーバ16-6と16-7がある図を示し、クライアント12とキャッシュサーバ16-6間の他に、キャッシュサーバ16-7とホームサーバ20間にも直接の経路が存在する。

しかし、変換が中間キャッシュサーバ16-9と16-10で実行されるとき、上述したように、この構成ではさらに一般的な構成を排除することはない。すなわち、図10で、要求または応答または両方申に、変換が中間キャッシュサーバ16-9と16-10でなされてもよい。

特定のホームサーバ20とクライアント12がキャッシュサーバ16を使用して通信をコミットされている限り、この方法は、別の形で帯域幅を節約する、透過的なドキュメント圧縮方式を許容する。特に、クライアント12とサーバ20間の経路に沿った最初のホップ後の、最初のリーフノードサーバ16-6は、クライアント12とホームサーバ20の両者に透過的な圧縮または暗号化を実装できる。

ドキュメントは、また、送出される前に、精細な検査とコード認証ができる。

本発明は、また、動的内容のドキュメントの効率的なキャッシュができ、クライアントに返送されるデータの最終結果は実行中にでも変化する。たとえば、Javaコードが埋め込まれているドキュメントには、このようなデータが含まれている。

キャッシュサーバ16は、ドキュメント用データをキャッシュするだけでなく、データ再生時にドキュメントを表示する方法を指定したプログラムをキャッシュして、このような動的内容のドキュメントの効率的な配布を達成できる。プロ

グラムがドキュメントを表示するときに、プログラムがクライアント12で通常実行される形式ならば、そのプログラムとデータは単にクライアント12に転送されるだけである。

しかし、そのプログラムが、クライアント12がホームサーバ20上で実行することを希望する形式であれば、キャッシュサーバ16はホームサーバの付加的レベルを実行し、そのプログラムも実行することでホームサーバと見せかける（代理の役割を果たす）。この時、キャッシュサーバ16がクライアント12と相互作用的セッション状態を維持して

、ホームサーバ20と完全に見せかける必要がある。

本発明は、また、本来、ホームサーバ20とクライアント12間のホップ数が少ないために、オーディオやビデオデータのファイルのような記憶メディアの配布に有効なものである

本発明によるいくつかの実施形態を示し説明したが、本発明がそれに限定されず、当業者には既知の多数の変更と修正が可能である。したがって、通常の技術を有する者には明白な変更と修正のすべてをカバーする以外は、ここで示し説明した詳細を制限することを望むものではない。

【図1】

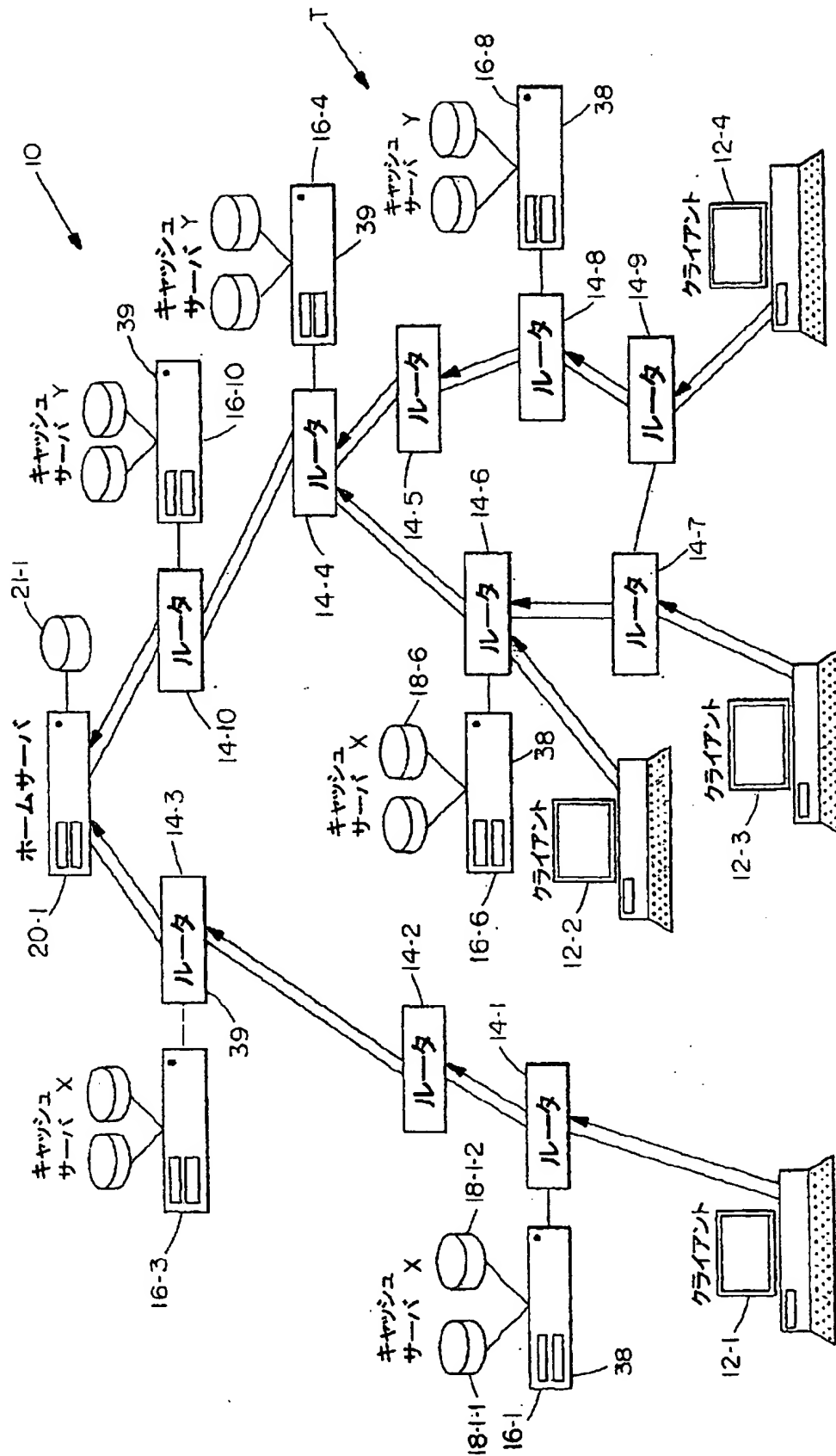


FIG. 1

【図2】

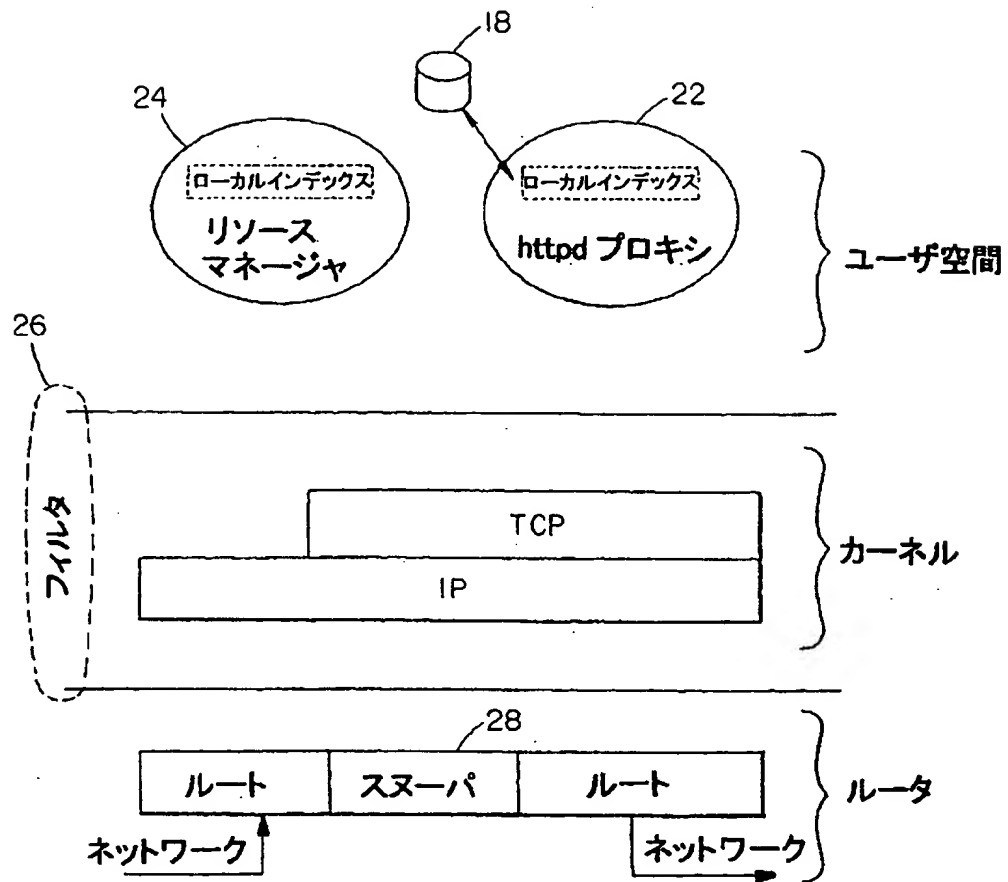


FIG. 2

【図3】

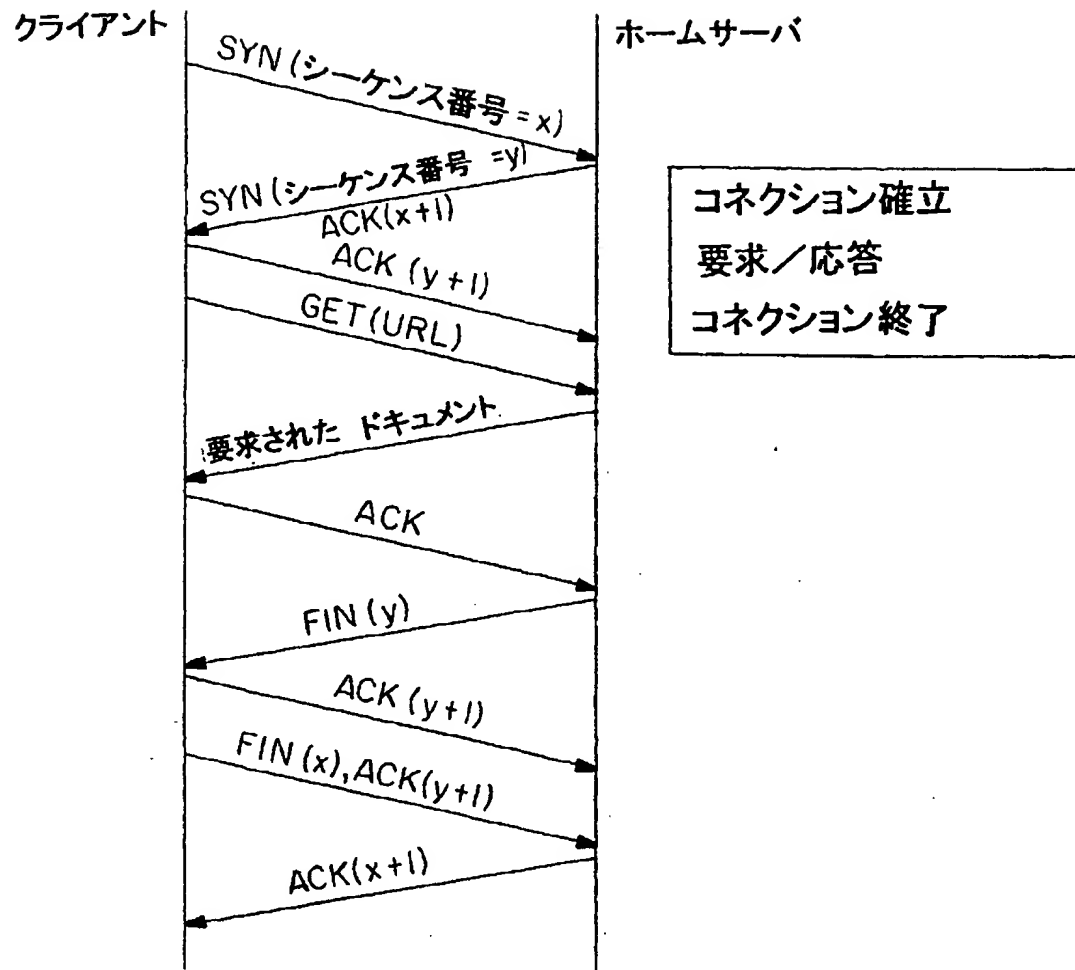


FIG. 3

【図4】

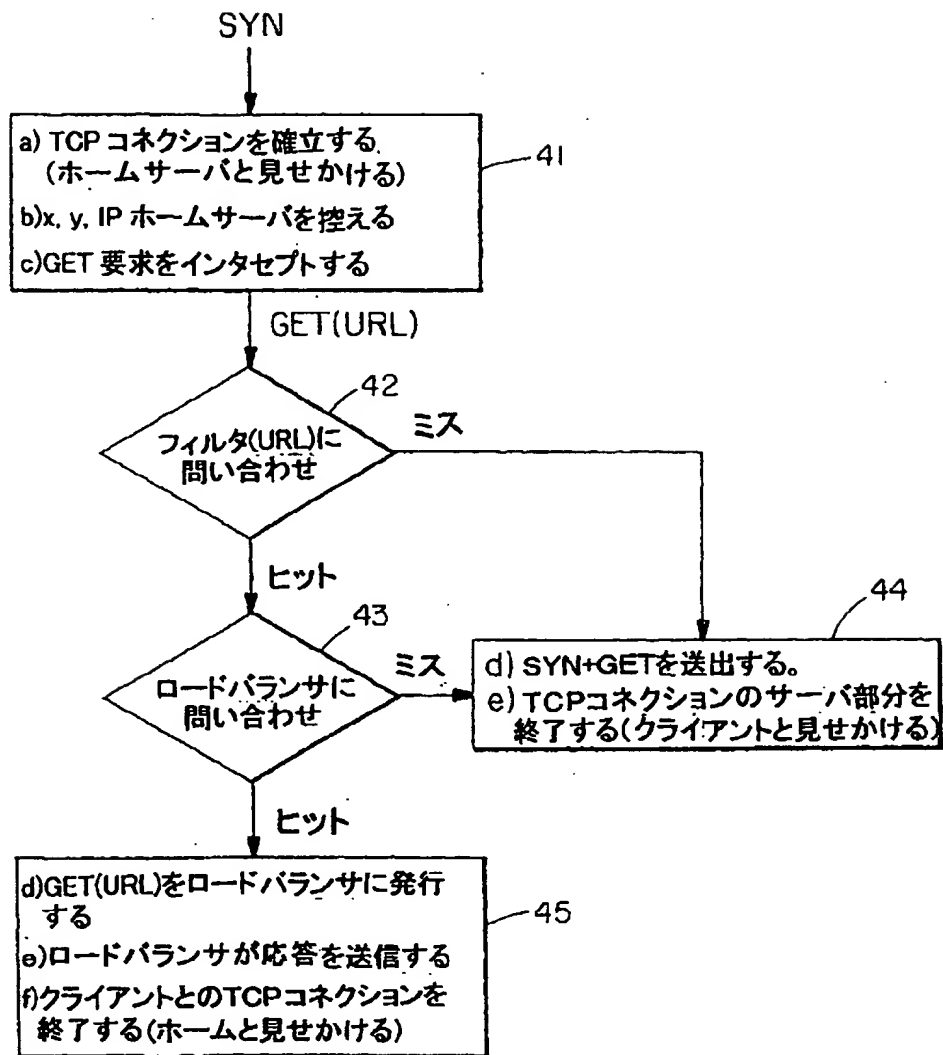


FIG. 4

【図 5】

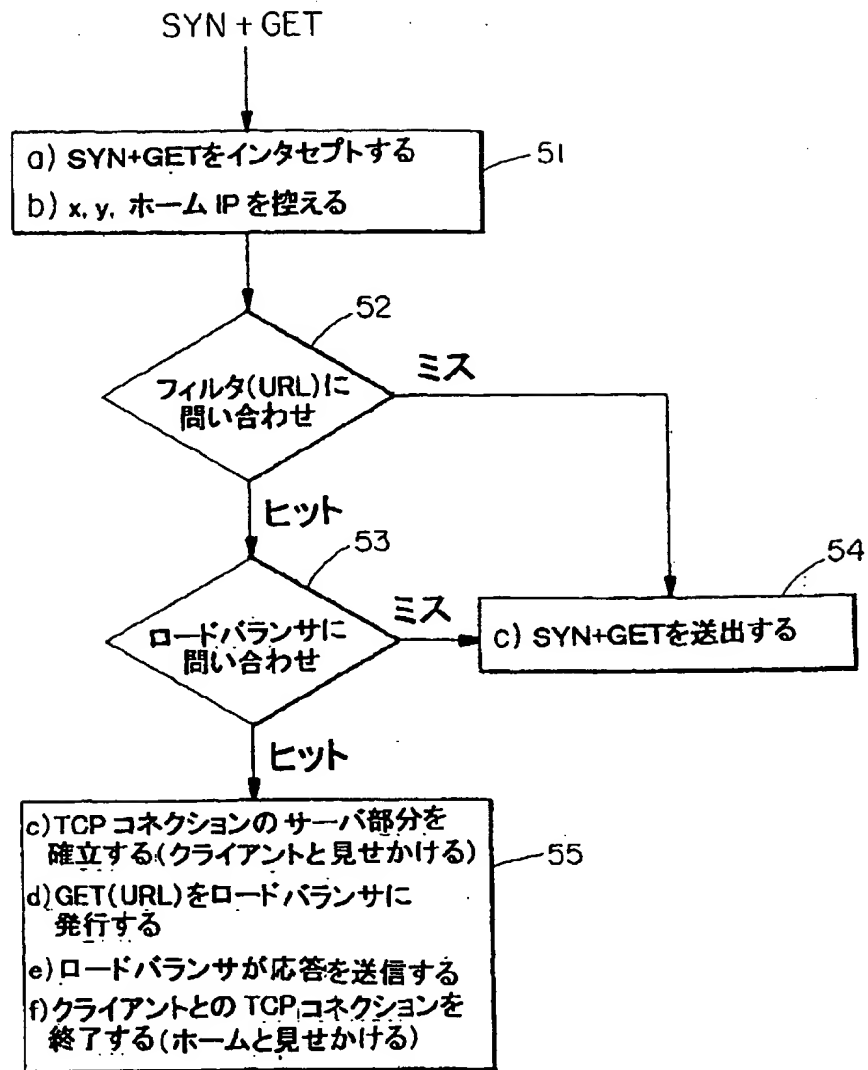


FIG. 5

【図6】

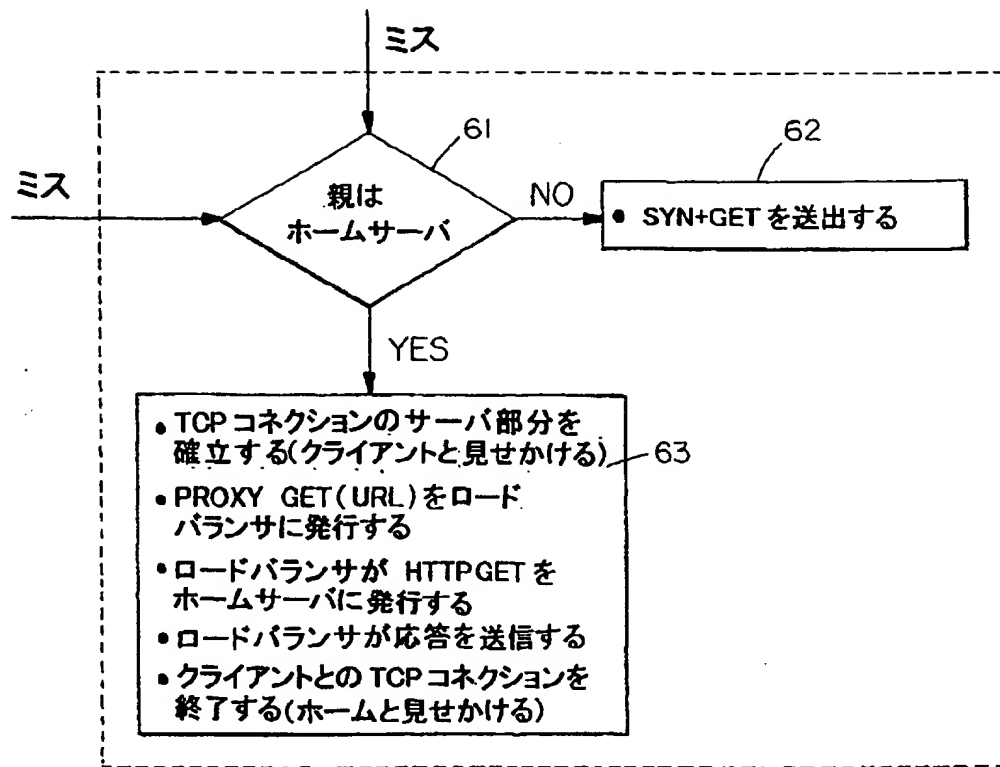


FIG. 6



【図7】

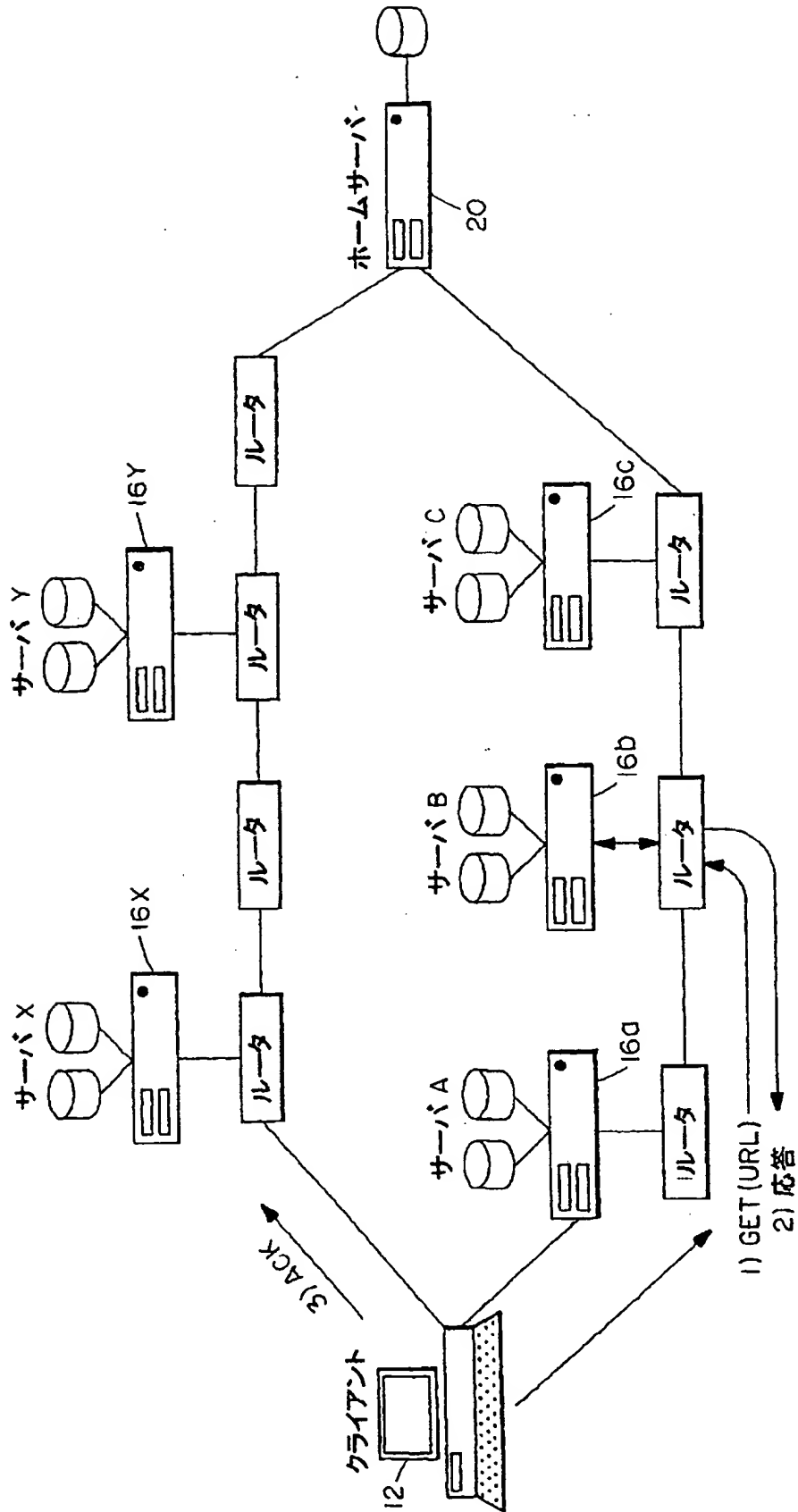


FIG. 7

【図8】

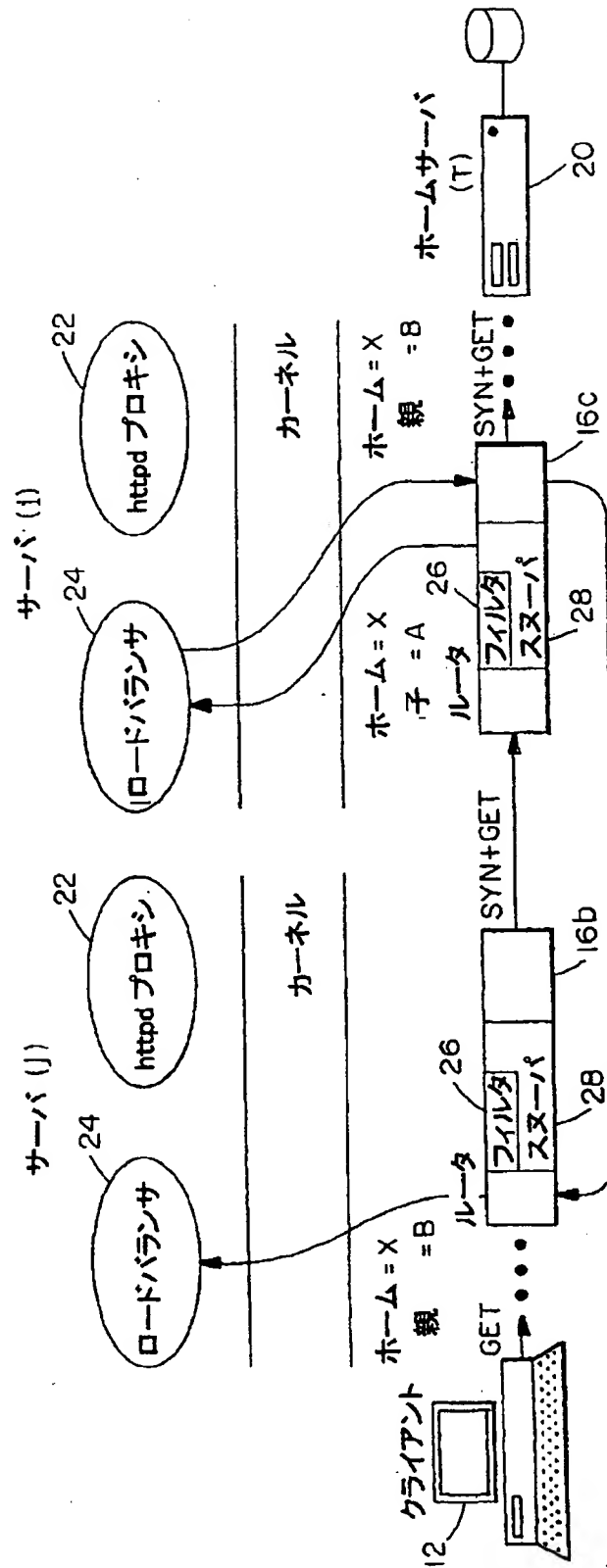


FIG. 8

【図9】

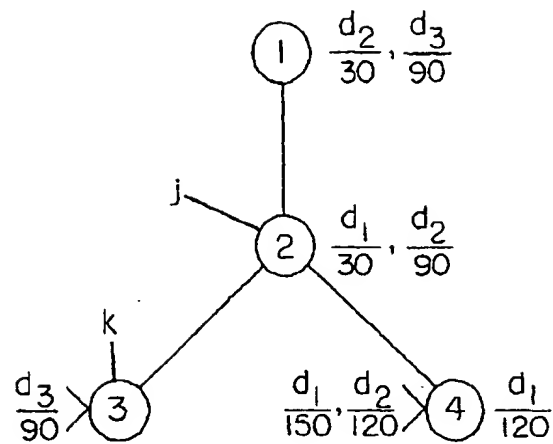


FIG. 9A

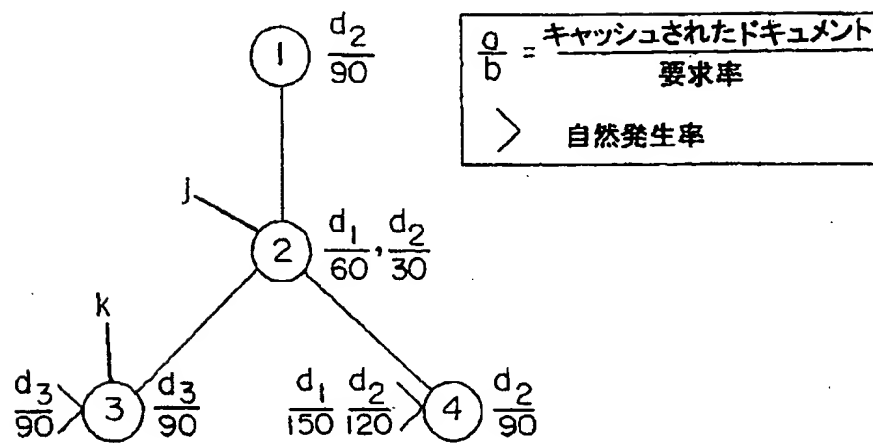


FIG. 9B

【図10】

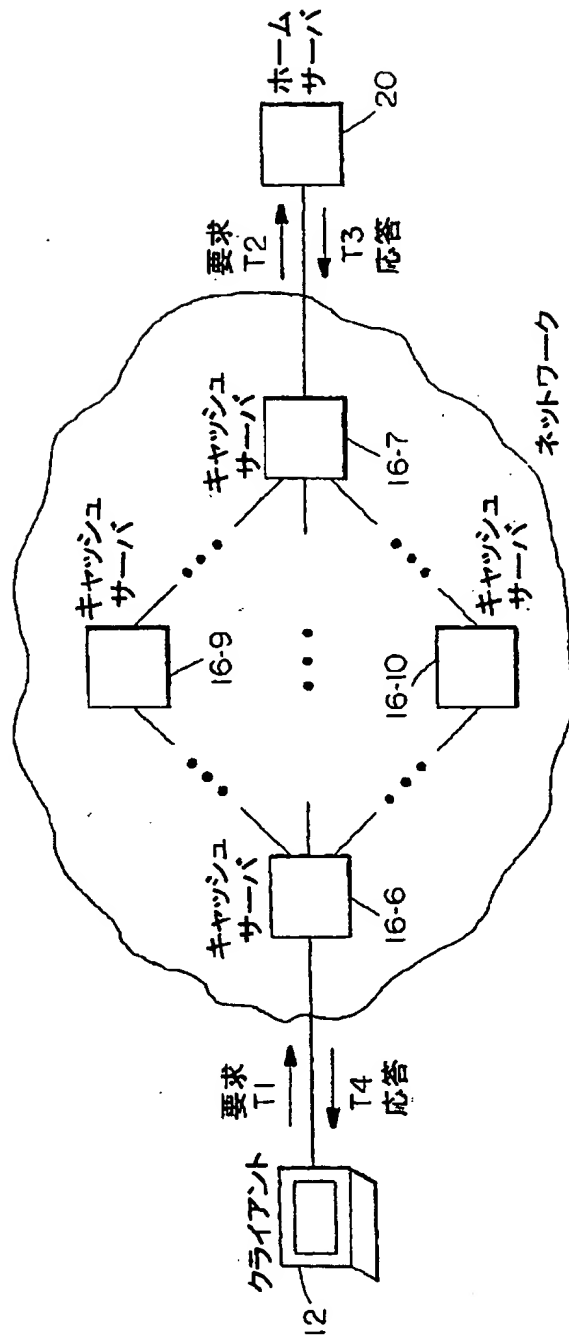


FIG. 10

【国際調査報告】

## INTERNATIONAL SEARCH REPORT

International Application No.

PCT/JP 98/09943

A. CLASSIFICATION OF SUBJECT MATTER  
IPC 6 G06F17/30 H04L29/06

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 6 G06F H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
	- / - -	

☒ Further documents are listed in the continuation of box C.☒ Patent family members are listed in annex.

## \* Special categories of cited documents :

- \*A\* document defining the general state of the art which is not considered to be of particular relevance
- \*E\* earlier document but published on or after the international filing date
- \*L\* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- \*O\* document referring to an oral disclosure, use, exhibition or other means
- \*P\* document published prior to the international filing date but later than the priority date claimed

- \*T\* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- \*X\* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- \*Y\* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
- \*Z\* document member of the same patent family

Date of the actual completion of the international search

25 August 1998

Date of mailing of the international search report

24. 11. 98

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel: (+31-70) 346-2040, Tx. 31 851 epo nl,  
Fax: (+31-70) 340-3016

Authorized officer

FOURNIER, C

## INTERNATIONAL SEARCH REPORT

International Application No.

PCT/US 98/09943

## C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Description of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	A. HEDDAYA & S. MIRDAD: "WebWave: Globally Load Balanced Fully Distributed Caching of Hot Published Documents" TECHNICAL REPORT BU-CS-96-024, BOSTON UNIVERSITY, CS DEPT, October 1996, XP002075420 printed on August 21st, 1998 and retrieved from Internet at: <a href="http://www.cs.bu.edu/techreports/abstracts/96-024">http://www.cs.bu.edu/techreports/abstracts/96-024</a> see abstract	1
P,X	-& HEDDAYA A ET AL: "WebWave: globally load balanced fully distributed caching of hot published documents" PROCEEDINGS OF THE 17TH INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS (CAT. NO.97CB36053), PROCEEDINGS OF 17TH INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS, BALTIMORE, MD, USA, 27-30 MAY 1997, ISBN 0-8186-7813-5, 1997, LOS ALAMITOS, CA, USA, IEEE COMPUT. SOC. PRESS, USA, pages 160-168, XP002075421	1-38, 43-62, 68,77
X	SHRIKUMAR H ET AL: "Thinternet: life at the end of a tether" COMPUTER NETWORKS AND ISDN SYSTEMS, vol. 27, no. 3, December 1994, page 375-385 XP004037970 see page 379, right-hand column, paragraph 3.2 - page 381, left-hand column, paragraph 3.3	1,2,43, 44,53,68
A	WO 93 24890 A (PITTS WILLIAM R) 9 December 1993 see page 15, line 1 - page 25, line 12 see page 9, line 9 - page 13, line 9	1-38, 40-77
A	BOLOT J ET AL: "Performance engineering of the World Wide Web: Application to dimensioning and cache design" COMPUTER NETWORKS AND ISDN SYSTEMS, vol. 28, no. 11, May 1996, page 1397-1405 XP004018237 see page 1403, right-hand column, line 1 - page 1405, right-hand column, paragraph 4	1,18
	---	
	-/--	

## INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 98/09943

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>BRAUN H ET AL: "Web traffic characterization: an assessment of the impact of caching documents from NCSA's web server" COMPUTER NETWORKS AND ISDN SYSTEMS, vol. 28, no. 1, December 1995, page 37-51 XP004001209 see page 49, right-hand column, paragraph 5.4 - page 50, right-hand column, line 20 -----</p>	<p>1,20,21, 53,63,67.</p>

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 98/09943

**Box I Observations where certain claims were found unsearchable (Continuation of Item 1 of first sheet)**

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:  
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☐ Claims Nos.:  
because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:
3. ☐ Claims Nos.:  
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

**Box II Observations where unity of invention is lacking (Continuation of Item 2 of first sheet)**

This International Searching Authority found multiple inventions in this International application, as follows:

Claims 1-38, 41-77: Method for distributed caching, prefetching and replication  
 Claim 39: Method for providing a secure communication path in a network of computers.  
 Claim 40: Method for reducing bandwidth communication path in a network of computers.

1. ☐ As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

1-38, 41-77

Remark on Protest

- ☐ The additional search fees were accompanied by the applicant's protest.
- ☐ No protest accompanied the payment of additional search fees.



## INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No.

PCT/US 98/09943

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9324890 A	09-12-1993	CA 2136727 A	09-12-1993
		AU 2227792 A	30-12-1993
		EP 0643853 A	22-03-1995
		JP 8500918 T	30-01-1996
		US 5611049 A	11-03-1997
-----			

## フロントページの続き

(51) Int. Cl. <sup>7</sup>	識別記号	F I	テーマコード* (参考)
H 0 4 L 12/58		H 0 4 L 11/20	1 0 1 Z
29/06			

(81) 指定国 EP(AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OA(BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG), AP(GH, GM, KE, LS, MW, SD, SZ, UG, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, GW, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW

(72) 発明者 ヘッダヤ・アブデルサラム・エー  
 アメリカ合衆国, マサチューセッツ州  
 02154 ウォルザム, スターンズ ヒル  
 ロード 901

(72) 発明者 イエーテス・ディビッド・ジェー  
 アメリカ合衆国, マサチューセッツ州  
 02062 ノーウッド, ビレッジ ロード  
 ウェスト 2809

## 【要約の続き】

装する。